



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrपाल Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645



Lt. Shree Chimanbhai Shukla

B.SC(I.T. & C.A) SEM-3(CPP)

Shree H.N.Shukla College Campus,
Street No. 2, Vaishali Nagar,
Nr. Amrapali Railway Crossing,
Raiya Road, Rajkot.
Ph. (0281)2440478, 2472590



Shree H.N.Shukla College Campus,
Street No. 3, Vaishali Nagar,
Nr. Amrapali Railway Crossing,
Raiya Road, Rajkot.
Ph. (0281)2471645

“C++”

CHAPTER-1

Sr No.	Topic
1.	POP
2.	OOP
3.	Benefits and Application of OOP
4.	What is c++? Application and features of c++
5.	Explain input and output operators in c++
6.	Structure of c++ Program
7.	Tokens in c++
8.	Reference Variable
9.	Data Types in c++
10.	Operator
11.	Expression
12.	Assignment Operator
13.	Implicit Conversion
14.	Operator Precedence and Associativity
15.	Control Structures
16.	Loop



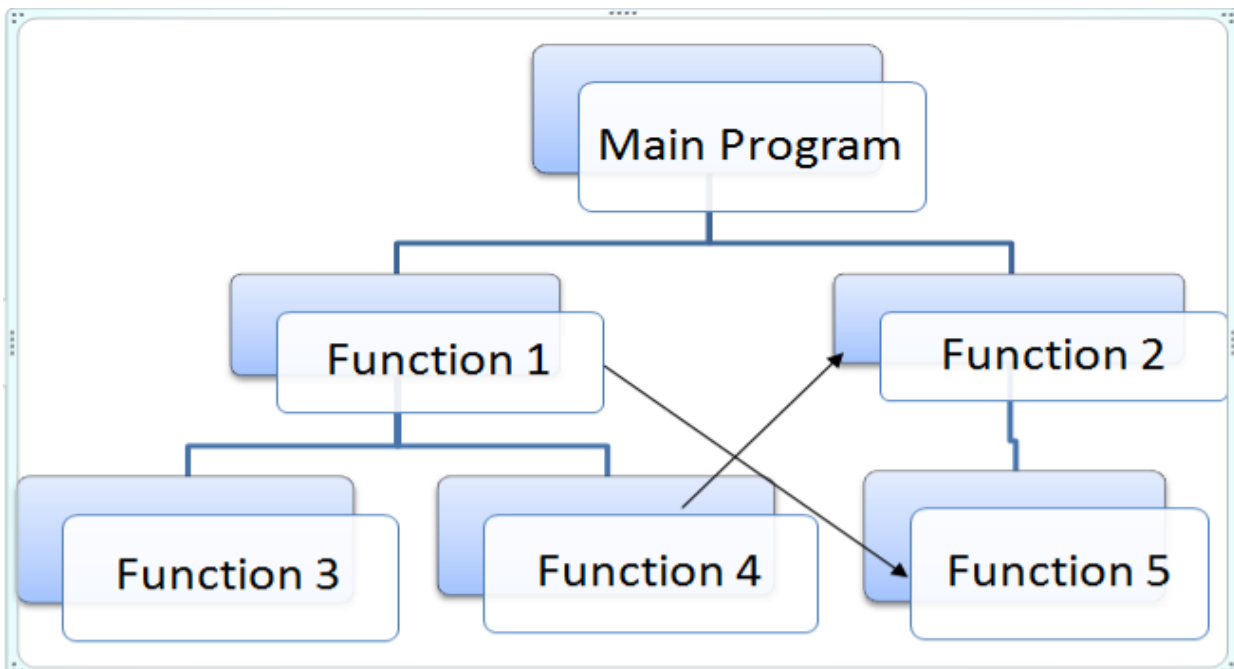
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: procedure-oriented programming (POP)

- **Procedural programming** uses a list of instructions to tell the computer what to do step-by-step.
- It based upon the concept of the **procedure** call.
- **Procedures**, also known as routines, or functions (not to be confused with mathematical functions), but similar to those used in functional **programming**.





SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- C Language is the example of POP.
- In POP, there is no security that is one program can access information of other program that is in POP, there is no concept of access modifier.

POP	OOP
1) POP stands for procedure oriented programming	1) OOP stands for Object Oriented Programming
2) In POP, there is no security of data.	2) In OOP, data is secured.
3) POP does not have concept of access modifier	3) OOP have concept of access modifier
4) C language is example of POP	4) C++ is the example of OOP
5) In POP, the program is divided in to functions	5) In OOP, the program is divided into objects

1 word Question Answer

Sr No.	Question	Answer
1.	POP Stands for?	procedure-oriented programming



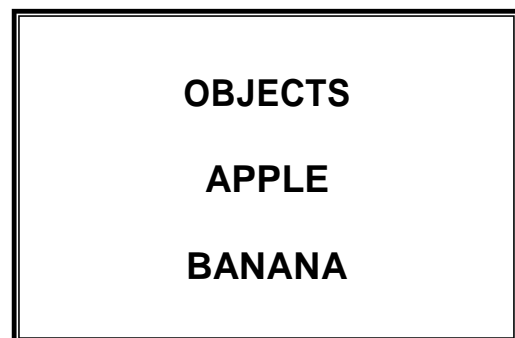
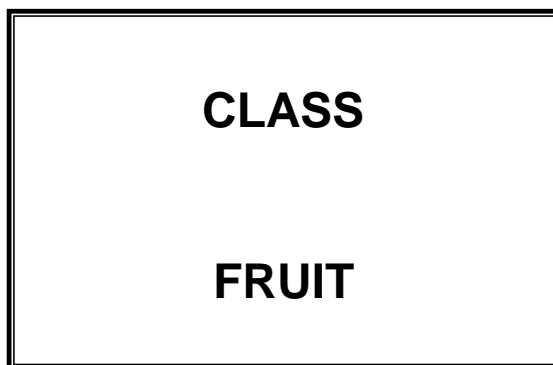
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: OOP

- OOP stands for Object-Oriented Programming.
- Classes and objects are the two main aspects of object-oriented programming.



- Class is the template for objects and object is the instance of the class.
- When the individual objects are created, they inherit all the variables and functions from the class.

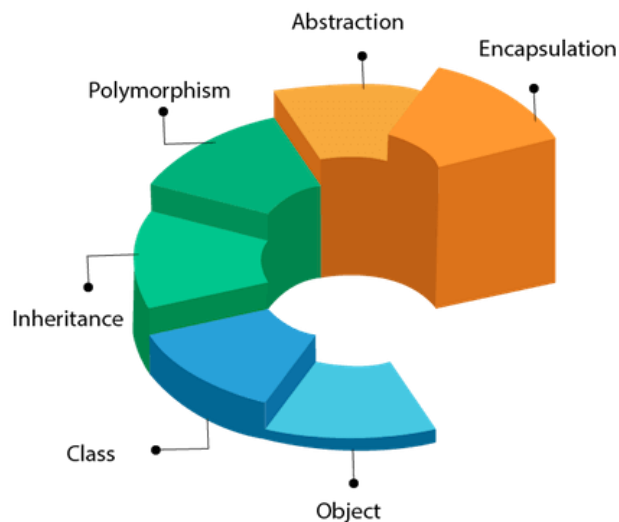


SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

OOPs (Object-Oriented Programming System)



Class:

- The building block of C++ that leads to Object-Oriented programming is a Class. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.
- A Class is a user-defined data-type which has data members and member functions.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax:

Class A

```
{  
  
    ...  
  
    ...  
  
}
```

Object:

- An Object is an identifiable entity with some characteristics and behavior.
- An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

Example:

Class person

```
{  
    char name[20];  
    int id;  
public:
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
void getdetails(){  
};  
  
int main()  
{  
    person p1; // p1 is a object  
}
```

Inheritance:

- Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.
- This is a very important concept of object-oriented programming since this feature helps to reduce the code size.

Polymorphism:

- Poly means multiple and morph means form (Multiple Forms)
- The ability to use an operator or function in different ways in other words giving different meaning or functions to the Operators or functions are called polymorphism.
- That is a single function or an operator functioning in many



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

ways different upon the usage is called polymorphism.

Abstraction:

- Data abstraction refers to, providing only essential information to the outside world and hiding their background details, i.e., to represent the needed information in program without presenting the details.
- For example, Outer lock of mobile phone, like it has display screen and keypad buttons to dial number

Encapsulation:

- Encapsulation is placing the data and the functions that work on that data in the same place.
- Example: Inside Mobile phone, how display screen and keypad are related to each other.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Difference between Abstraction and Encapsulation

Abstraction	Encapsulation
1) Abstraction is used to hide unwanted data and give relevant data.	1) Encapsulation means hiding the data and code in single unit to protect the data from outside world.
2) Abstraction Solves the problem at design level	2) Encapsulation solves the problem at implementation level.
3)	
4) Example: Outer lock of mobile phone, like it has display screen and keypad buttons to dial number. (Which means only the needed information to unlock the phone is provided.)	3) Example: Inside Mobile phone, how display screen and keypad are related to each other.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

1 word Question Answer

Sr No.	Question	Answer
1. and are the components of OOP	Class and object
2.is the instance of class	Object
3.	OOP stands for	Object Oriented Programming
4.concept uses parent child concept	Inheritance
5.	The process of representing the same work in different form is known as....	Polymorphism

Topic: Benefits and Applications of OOP

Benefits of OOP:

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the C++ code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorter development time



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Applications of OOP:

- User interface design such as windows, menu.
- Real Time Systems
- Simulation and Modeling
- Object oriented databases
- AI and Expert System
- Neural Networks and parallel programming
- Decision support and office automation systems etc.

Topic: What is C++? Explain features and applications of c++

- C++ is a high level object oriented programming language that helps programmers write fast, portable programs.
- C++ is the extension of C Language.
- It was developed by **Bjarne Stroustrup**

Features of C++:

1) Object Oriented:

C++ is object oriented programming language and supports all the concepts of OOP that is inheritance, Polymorphism etc.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- 2) Rich Library support:
Through c++ standard template library (STL) many functions are available that help in quickly writing code.
- 3) Speed:
The compilation as well as execution time of c++ program is much faster than most other general purpose programming languages.
- 4) Pointer Support: C++ also supports pointers which are widely used in programming and are often not available in several programming languages.

Application of C++:

- 1) Operating System
- 2) Browsers
- 3) Libraries
- 4) Graphics
- 5) Banking Applications
- 6) Database

Topic: Explain input and output operators in c++

- C++ is able to input and output the built-in data types using the stream extraction operator >> and the stream insertion operator <<.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- The stream insertion and stream extraction operators also can be overloaded to perform input and output for user-defined types like an object.
- cout object is used with insertion operator. <<
- cin object is used with extraction operator >>

Example:

cout<<"hello world"
↓ ↓ ↓
Object Insertion Operator Message

cin>> Number1
↓ ↓ ↓
Object Extraction Operator Value of number you want to input



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

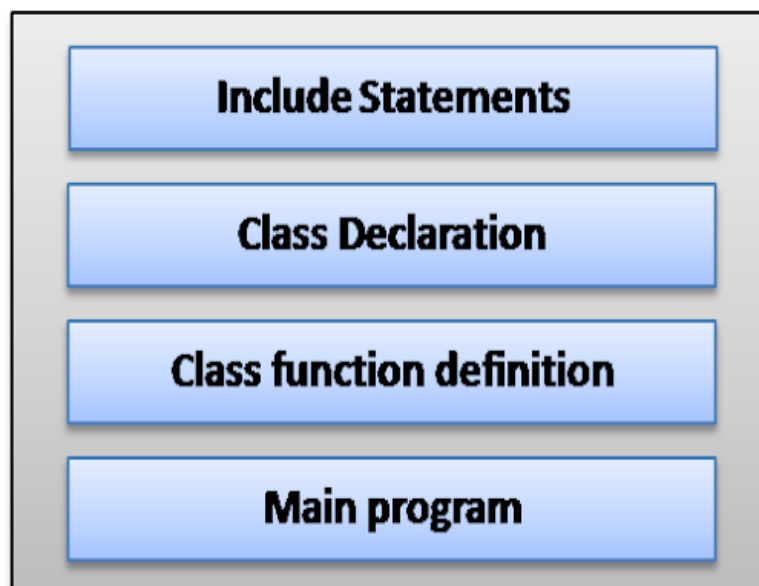
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

1 word Question Answer

Sr No.	Question	Answer
1. operator is known as insertion operator	<<
2.operator is known as extraction operator	>>
3.object is used with insertion operator	cout
4.object is used with extraction operator	cin

Topic: Explain structure of c++ program.



“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

1) Include statements:

- Include statements are also known as pre-processor directives.
- This statement is used to include header files in the program.
- Example: #include<iostream.h>

2) Class Declaration:

- As c++ is object oriented programming language, it uses concept of class and object.
- So, second section that is after header file, you can create the class according to your program requirement.
- Example:

```
Class Hello
```

```
{
```

```
....
```

```
...
```

```
};
```

Hello is name of class

3) class Function Definition/Member Function Definition:

- Class Function Definition means the body of the functions which are used inside the class.
- Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
Class Hello
{
    void print()
    {
        ...
        ...
    }
};
```

**print() is class function
definition**

4) Main Program:

- After the declaration of class and function, we have to declare main()
- Without main(), program does not get execute.

Example:

```
Class Hello
{
    void print()
    {
        ...
        ...
    }
};
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
void main()  
{  
    Hello h;  
    h.print();  
    getch();  
}
```

main()

Topic: Explain tokens in c++

Ans:

- A token is the smallest element of a program that is meaningful to the compiler.
- Tokens can be classified as follows:
 1. Keywords
 2. Identifiers
 3. Constants
 4. Strings
 5. Special Symbols
 6. Operators

1) Keywords:

- Keywords are pre-defined or reserved words in a programming language. Each keyword is meant to perform a specific function in a program.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- Since keywords are referred names for a compiler, they can't be used as variable names because by doing so, we are trying to assign a new meaning to the keyword which is not allowed.
- In **C++**, there are **31** additional keywords other than **C** Keywords they are:

```
asm      bool      catch      class
const_cast  delete  dynamic_cast  explicit
export   false     friend     inline
mutable  namespace  new        operator
private  protected  public     reinterpret_cast
static_cast  template  this       throw
true     try        typeid     typename
using    virtual   wchar_t
```

2) Identifiers:

- Identifiers are used as the general terminology for naming of variables, functions and arrays.
- These are user defined names consisting of arbitrarily long sequence of letters and digits with either a letter or the underscore(_) as a first character



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- There are certain rules that should be followed while naming C identifiers:

- 1) They must begin with letter or underscore().
- 2) They must consist of only letters, digits, or underscore, No other special character is allowed
- 3) It should not be a keyword
- 4) It must not contain white space
- 5) It should be up to 31 characters long as only first 31 characters are significant.

Example:

```
void main()  
{  
    int a;  
}
```

In the above example, there are 2 identifiers

- 1) main: Method name
- 2) a: variable name

3) Constant:

- Constants are also like normal variables.
- But, only difference is, their values cannot be modified by the program once they are defined.



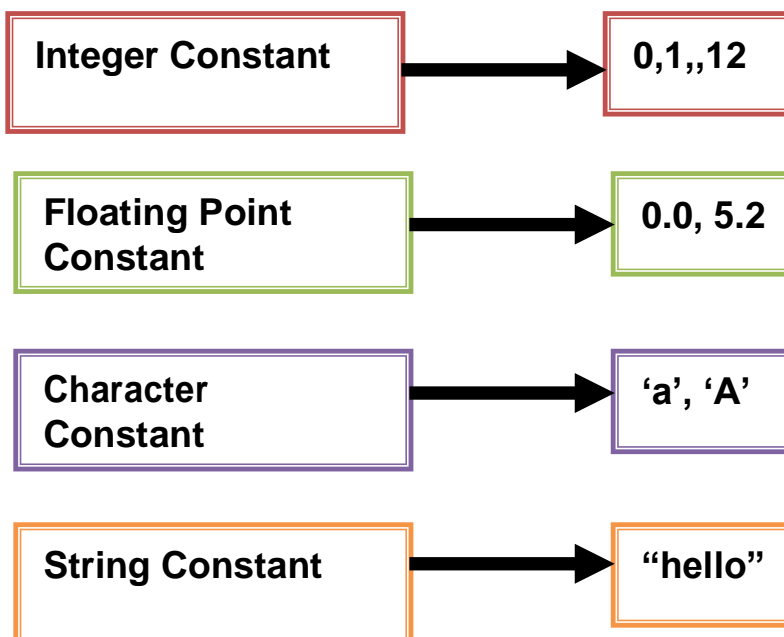
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- Constants refer to fixed values. They are also called as literals.

Types of Constant:



Note:

```
char string[20]="hello how are you"  
char string[]="hello how are you"
```

- The difference in the above declaration is that when we write `char string[20]` then 20bytes of memory space is allocated to



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

the string and when we write char string[] then memory space is allocated according to the requirement

4) Special Symbols:

These are the following types of symbols:

- ❖ **Brackets[]**: Opening and closing brackets are used as array element reference. These indicate single and multidimensional subscripts.
- ❖ **Parentheses()**: These special symbols are used to indicate function calls and function parameters.
- ❖ **Braces{}**: These opening and ending curly braces marks the start and end of a block of code containing more than one executable statement.
- ❖ **comma (,)**: It is used to separate more than one statements like for separating parameters in function calls.
- ❖ **semi colon :** It is an operator that essentially invokes something called an initialization list.
- ❖ **asterisk (*)**: It is used to create pointer variable.
- ❖ **assignment operator**: It is used to assign values.
- ❖ **Pre processor(#)**: The preprocessor is a macro processor that is used automatically by the compiler to transform your program before actual compilation.



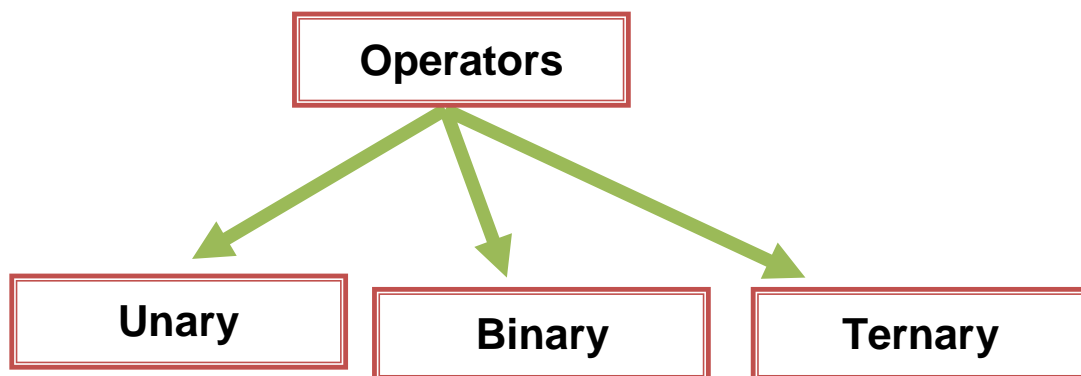
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

5) Operators:

- Operators are symbols that triggers an action when applied to variables and other objects.
- The data items on which operators act upon are called operands.



Unary Operator:

- Those operators that require only single operand to act upon are known as unary operators.
- For Example increment and decrement operators



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Binary Operator:

- Those operators that require two operands to act upon are called binary operators.
- Binary operators are classified into :
 - 1) Arithmetic operators
 - 2) Relational Operators
 - 3) Logical Operators
 - 4) Assignment Operators
 - 5) Conditional Operators
 - 6) Bitwise Operators

Ternary Operator:

- This operator requires three operands to act upon. For Example Conditional operator (?:).

Topic: Explain reference variable

Ans:

- A reference variable provides an alternative (alias) for previously defined variable.

Example:

```
int amount=100;
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
int &total =amount;  
cout<<amount<<total;
```

In the above example, **&total** is the reference variable.

• **Features of reference variable:**

- 1) A reference variable must always be initialized.
int &j // Not valid
- 2) Once a reference variable has been defined to refer to a particular variable, it cannot refer to any other variable.
- 3) A variable can have multiple references.
- 4) A reference of pointer can also be created.

Topic: Explain data types in c++

Ans:

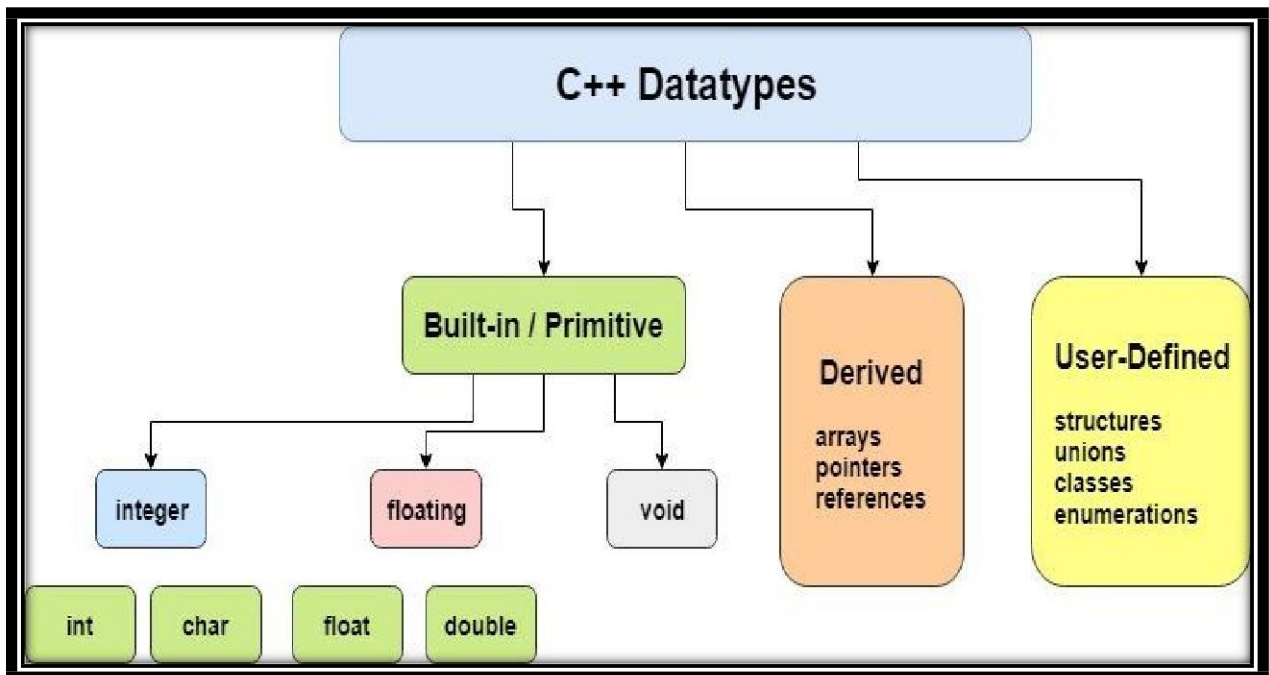
- Data types are used to tell the variables the type of data it can store.
- Whenever a variable is defined in C++, the compiler allocates some memory for that variable based on the data-type with which it is declared.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645



1) Built-in data types:

The data types which are pre-defined that is which are not created by the users is known as Built-in data types.

Example: integer, float, char etc.

2) User Defined Data Types:

The data types which are created by the user itself according to his requirements is known as user defined data types.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

- 1) **Class:** It is new user defined data type in c++ which is the collection of variables, methods etc.

Class Hello

```
{  
    void display()  
    {  
        Response.Write("Hello");  
    }  
}
```

- 2) **Structure:** It is the collection of variables that may or may not have same data type.

```
struct student  
{  
    int roll;  
    char name[20];  
};
```

- 3) **Union:** It is same as structure except in terms of memory location. In Union, the variable having largest data type, memory of that data type will be allocated to the entire union.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Union Student

```
{  
    int roll;  
    char name[20];  
};
```

4) Enumeration: This data type is used to hold multiple constant values.

```
enum week={"Sunday", "Monday", ..... }
```

3) Derived Data Types:

The data types which are derived from built in data types is known as derived data types.

Example:

1) Function: This is the derived data type in which user can code according to his requirement

```
void add()  
{  
    int a=5,b=5,sum;  
    sum=a+b;  
    cout<<"Sum is"<<sum;  
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

2) **Array:** It is the collection of elements with same data types.

```
int a[5];
```

3) **Pointer:** It is also the derived data type which is used to store the address of another variable,

```
int a=10;  
int *p;  
p=&a;
```

Topic: Explain Operators in c++

Ans:

1) Scope Resolution Operator(: :)

- This operator is used to display global variables which have same name as local variable.

Example:

```
int a=10; //Global Variable  
void main()  
{  
    int a=10; //Local Variable  
    cout<<"Local Variable"<<a;
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
cout<<"Global Variable"<<::a;  
getch();  
}
```

2) Member Dereferencing Operator:

C++ permits to access the class members through pointers for that there is set of three pointer to member operator,

::* Declare a pointer to member of class

.* Access the member using object name

->* To display values of pointer using object name

3) Memory Management Operator:

- C++ supports dynamic memory management using keywords new and delete.

new: This operator is used to allocate memory dynamically to pointer variable

Syntax: datatype *Pointervariable=new datatype;

Example: int *p=new int;



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

delete: This operator is used to free the memory that is allocated to pointer variable

Syntax: delete pointervariablename;

Example: delete p

4) Manipulators:

These are the operators that are used to format the data display. There are most commonly two manipulators:

- 1) endl
- 2) setw

endl: This manipulator is used in an output statement to enter the text or data in new line

Example:

```
cout<<"Hello"<<endl;  
cout<<"Hi"
```

setw: This manipulator is used to set the width of output.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
cout<<setw(10)<<"Hello"
```

5) Type Cast Operator:

This operator is used to explicitly typecast the data type.
It is used to convert one data type into another data type.

Syntax: (type)expression;

Example:

```
main() {  
    double a = 21.09399;  
    float b = 10.20;  
    int c ;  
  
    c = (int) a;  
    cout << "Line 1 - Value of (int)a is :" << c << endl ;  
  
    c = (int) b;  
    cout << "Line 2 - Value of (int)b is :" << c << endl ;  
  
    return 0;  
}
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on Expression

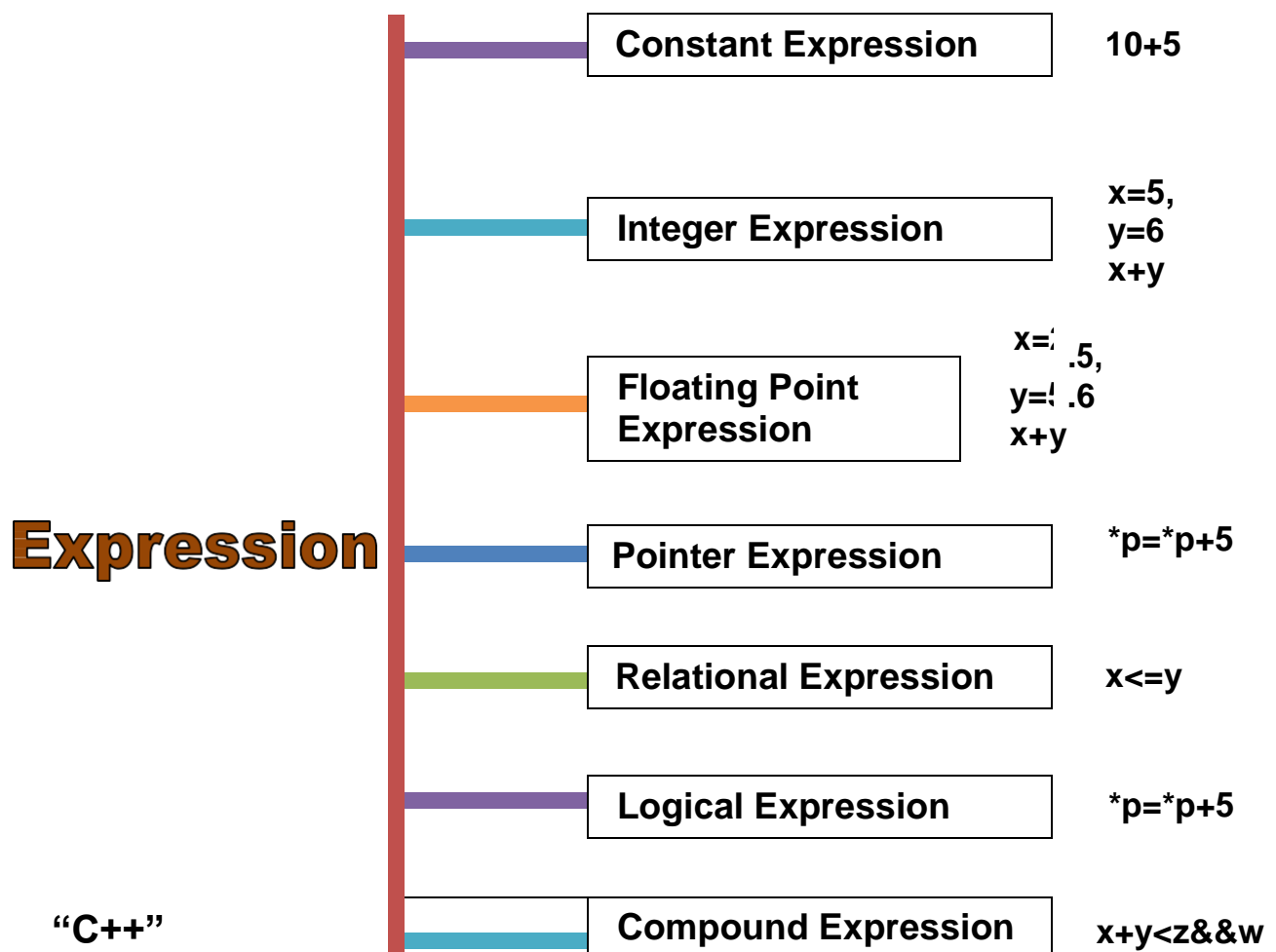
Ans:

- Expression means combination of variables, constant or symbols or operators.

Example

$$x+y=5;$$

- Following are the types of expression.





SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on Assignment Operator

Ans:

- This operator is used to assign the value to a variable.
- = symbol is known as assignment operator.

Syntax: variable=expression

Example: a=5

- Following are the types of assignment operator.

1) Chained Assignment:

- In this type of assignment, there are more than 1 variables and chained assignment cannot be initialized at the time of declaration.
- Example:

`x=y=10;`

2) Embedded Assignment:

- In this type of assignment, the statement is joined or embed in the assignment.
- Example:

`x=(y=50)+10`



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

3) Compound Assignment:

- Compound assignment is the combination of assignment operator with a binary operator.
- **It is also known as shorthand operator.**
- Example:

$x=x+10$

$x+=10$

Topic: Write a short note on Implicit Conversion

Ans:

- Whenever there are more than 1 data types in the expression, C++ performs the conversion automatically. This process is known as implicit or automatic conversion.
- Implicit conversion is also known as **silent conversion.**
- Rules for implicit conversion are:

Operand1	Operand 2	Result
char	int	int
int	long	long
int	float	float
int	double	double
double	float	double



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
int a;  
float b;
```

- In the above expression, variable a has int datatype and variable b has float data type. Among that 2 data types, float is larger data type than integer so as per the rule, int is converted into float automatically.

Topic: Write a short note on Operator Precedence and associativity.

Ans:

When in the same expression, we have more than one operators then which operator should be solved first? or which operator should be given priority is the matter of consideration.

- Operator precedence means the operator having higher precedence than other operators.
- Example:

$$a+b*c$$

- In the above example, there are 2 operators in the expression, that is + and *..but the operator * have higher precedence than + so we will solve $b*c$ first and then result of $b*c$ is added to a.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

<i>Operator</i>	<i>Description</i>	<i>Associativity</i>
<code>()</code> <code>[]</code> <code>.</code> <code>-></code> <code>++ --</code>	Parentheses or function call Brackets or array subscript Dot or Member selection operator Arrow operator Postfix increment/decrement	left to right
<code>++ --</code> <code>+ -</code> <code>! ~</code> <code>(type)</code> <code>*</code> <code>&</code> <code>sizeof</code>	Prefix increment/decrement Unary plus and minus not operator and bitwise complement type cast Indirection or dereference operator Address of operator Determine size in bytes	right to left
<code>* / %</code>	Multiplication, division and modulus	left to right
<code>+ -</code>	Addition and subtraction	left to right
<code><< >></code>	Bitwise left shift and right shift	left to right
<code>< <=</code> <code>> >=</code>	relational less than/less than equal to relational greater than/greater than or equal to	left to right
<code>== !=</code>	Relational equal to or not equal to	left to right
<code>&&</code>	Bitwise AND	left to right
<code>^</code>	Bitwise exclusive OR	left to right
<code> </code>	Bitwise inclusive OR	left to right
<code>&&</code>	Logical AND	left to right
<code> </code>	Logical OR	left to right
<code>? :</code>	Ternary operator	right to left
<code>=</code> <code>+= -=</code> <code>*= /=</code> <code>%= &=</code> <code>^= =</code> <code><<= >>=</code>	Assignment operator Addition/subtraction assignment Multiplication/division assignment Modulus and bitwise assignment Bitwise exclusive/inclusive OR assignment	right to left
<code>,</code>	comma operator	left to right



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- In the expression, if there are different operators which have same precedence then at that time, associativity of operators must be considered.
- Associativity of operators have 2 values:
 - left to right
 - right to left

Example:

$a+b-c$

In the above example, + and – both have same precedence. So in such case, associativity should be taken in to consideration. The associativity is left to right so we will solve + first and then -

Topic: Write a short note on Control Structures

Ans:

- Control statements define the direction or flow in which execution of a program should take place.
- There are 4 types of control structures:
 - if
 - if-else
 - Nested if
 - Else if Ladder
 - Switch case



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

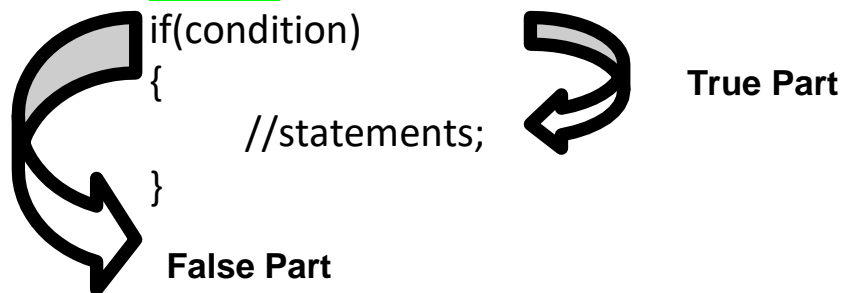
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

1) if:

Description: This control structure is used when we have only 1 condition and only true part is taken into consideration.

Syntax:



Example:

```
if(i==5)
{
    cout<<"Hello";
}
```

2) if-else:

Description: In this type of control structure, if condition is true then statement 1 is executed and if the condition is false then statement 2 is executed.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax:

```
if(condition)
{
    //statement 1;
}
else
{
    // statement 2;
}
```

False Part

True Part

Example:

```
if(i==5)
{
    cout<<"Hello";
}
else
{
    cout<<"Hi";
}
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

3) Nested if:

Description:

- Nested if means one if inside another if.
- Nested if is used when you have more than 1 condition.
- In nested if, there are 2 if-else that is inner if-else and outer if-else

Syntax:

```
if(condition 1)
{
    if(condition 2)
    {
        //statement 1;
    }
    else
    {
        //Statement 2;
    }
}
else
{
    if(condition 3)
    {
        // statement 3
    }
}
```

False Part

True Part



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
    else
    {
        //statement 4;
    }
}
```

Example:

```
int a=10,b=5,c=2;
if(a>b)
{
    if(a<c)
    {
        cout<<"a is max";
    }
    else
    {
        cout<<"c is max";
    }
}
else
{
    if(b>c)
        cout<<"b is max";
    else
        cout<<" c is max";
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

4) Else-if ladder:

Description:

- This decision control structure is also used when you work with more than 1 condition.
- if-elseif-else if-else if-else is known as else-if ladder.
- In else-if ladder when all the conditions are false then else part is executed.

Syntax:

```
if(condition 1)
{
    //Statement 1;
}
else if(condition 2)
{
    //Statement 2;
}
else if(condition 3)
{
    //Statement 3;
}
....
else
{
    //Statement 4;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
int a=10,b=5,c=2;
if(a>b && a>c)
cout<<"a is max";
else if(b>c && b>a)
cout<<"b is max";
else
cout<<"c is max";
```

5) switch case:

Description:

- Switch case is used when you have multiple options.
- In switch case, value of variable is passed and is compared with the different cases. The value which matches the case, that case is executed.
- break keyword is used to exit from the particular case.

Syntax:

```
switch(variable)
{
    case 1: statement 1;
           break;
    case 2: statement 2;
           break;
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
        default: statement 3;  
                break;  
    }
```

Example:

```
int a;  
switch(a)  
{  
    case 1: cout<<"Hello";  
            break;  
    case 2: cout<<"Hi";  
            break;  
    case 3: cout<<"How are you";  
            break;  
    default: cout<<"Invalid";  
            break;  
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: What is loop? List out looping structures and explain in detail

Ans:

- When same work is to be done multiple times, loop is used.
- There are 2 types of loop:
 - Entry controlled loop
 - Exit controlled loop

1) Entry Controlled loop:

- Entry controlled loop means the condition is checked at starting of the loop.
- There are 2 types of entry controlled loop:
 - 1) for loop
 - 2) While loop

1) for loop:

- This loop contains 3 parts: Initialization, Condition Checking, and Increment/decrement.

Syntax:

```
for(initialization; condition; increment/decrement)
{
    Statements;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
for(i=1;i<=5;i++)  
{  
    cout<<"Hello";  
}
```

2) while loop:

- This loop also contains 3 parts: Initialization, Condition Checking, and Increment/decrement.

Syntax:

```
Initialization;  
while(condition)  
{  
    statements;  
    increment/decrement;  
}
```

Example:

```
int i=1;  
while(i<=5)  
{  
    cout<<"Hello";  
    i++;  
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
int i=1;
while(i<=5)
{
    cout<<"Hello";
    i++;
}
```

2) Exit Controlled loop:

- Exit controlled loop means the condition is checked at ending of the loop.
- Do-while loop is example of exit controlled loop

Syntax:

```
Initialization;
do
{
    statements;
    increment/decrement;
}while(condition);
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
int i=1;  
do  
{  
    cout<<"Hello";  
    i++;  
}while(i<=5);
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

SUBJECT: C++

CHAPTER-2:

Sr No.	Topic
1.	What is Function
2.	Components of Function
3.	main()
4.	Call by value and call by reference
5.	Return by reference
6.	Inline Function
7.	Default Arguments
8.	Const Arguments
9.	Function Overloading
10.	C Structure Revisited
11.	Local Class
12.	Nested Class
13.	Member Function inside and outside the class
14.	Nesting of Member Function
15.	Private Member Function
16.	Array within class
17.	Memory allocation of objects
18.	Static Data Member & Static Member Function
19.	Array of Objects
20.	Friend Function



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: What is Function?

Ans:

- Functions are of 2 types:
 - 1) Inbuilt functions
 - 2) User Define Function

1) Inbuilt Functions:

- The functions which are available by default to use that is the functions which TC provides are known as inbuilt functions.
- Example: clrscr(), cout(), cin()

2) User Define Functions:

- The functions which are created by the user itself is known as UDF.

Sr No.	Question	Answer
1.	The functions which are available by default is known as?	Inbuilt Function
2.	The functions which are created by user is known as?	User define function



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Components of Function:

Ans:

The function has following components related to it

1) Function Declaration/Function Prototype:

- The function declaration tells the compiler about the parameter passed to the function and type of value returned by the function in advance.

Syntax:

```
returntype functionname(datatype, data type);
```

2) Function Definition:

- The function definition defines the body or the code of the function

Syntax:

```
returntype functionname(datatype, datatype)  
{  
    //body;  
}
```

3) Function Call:

- A function can start its execution when a call to function is made.
- A function call is specified by the function name followed by the arguments enclosed in the parentheses () and terminated by semicolon



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax:

functionname(arguments);

Example:

```
#include<iostream.h>
#include<conio.h>

void add(int,int);      //Function Declaration

void main()
{
    int a,b;
    clrscr();
    cout<<"Enter a";
    cin>>a;
    cout<<"Enter b";
    cin>>b;
    add(a,b);          //Function Call
    getch();
}

void add(int x, int y)  //Function Definition
{
    int z;
    z=x+y;
    cout<<"Answer is"<<z;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	There arecomponents of function	3
2.defines the body or the code of the function	Function Definition
3.	Any user defined function cannot be executed without calling that function (T/F)	True
4.	Function declaration is done after.....	Header files
5.	Does function declaration includes semicolon? (Y/N)	Yes

Topic: main()

Ans:

- main() is starting point or entry point of c++ program.
- main() also can have arguments or parameters that are known as command line arguments.



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax:

```
void main(int argc, char *argv[ ])  
{  
  
    //statements;  
  
}
```

- In the above syntax, void is the return type of main() that is main() will not return any value.
- argc: Indicates total number of arguments.
- argv[] is the character array which points to argument passed in command prompt.

argv[0]=First argument which is **Name of the program**
argv[1]= Second argument
argv[2]=Third argument
....
....

Example:

```
//Program to print hello using cmd (Command line arguments)
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
void main(int argc, char *argv[])
```

```
{
```

```
    cout<<argv[1];
```

```
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Steps:

- 1) Write the program
- 2) Compile your program
- 3) Run program
- 4) Go to file menu-> Dos Shell
- 5) Change directory
cd..
c:\TC\TURBOC
- 6) Go to source
cd source
- 7) c:TC\TURBOC\SOURCE cmda hello

program name

Sr No.	Question	Answer
1.is the starting point of program	main()
2.	We can pass arguments to main() using	cmd (Command Line Arguments)
3.	argv[0] always indicates	First argument: Program name



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain call by value and call by reference

Ans:

Call By Value:

- In call by value, the value of the variable is passed as arguments.
- In call by value, any changes made in the formal arguments are not reflected back to the actual arguments.

Note:

Formal Arguments:

The arguments which are passed in function definition are known as formal arguments

Actual Arguments:

The arguments which are passed in function call are known as actual arguments.

Example:

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>

void swap(int,int);           //Function Declaration

void main()
{
    int x,y;
    clrscr();
    cout<<"Enter x";
    cin>>x;
    cout<<"Enter y";
    cin>>y;
    swap(x,y);               //Function Call
    getch();
}

void swap(int a,int b)       //Function Definition
{
    int c;
    c=a;
    a=b;
    b=c;
    cout<<a;
    cout<<b;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES **(Affiliated to Saurashtra University and G.T.U)**

**2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478**

**3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645**

Call By Reference:

- In call by reference, the address of the variable is passed as arguments.
- In call by reference, any changes made in the formal arguments are reflected back to the actual arguments.

Call By Value	Call By Reference
1) In call by value, value of the variable is passed as arguments	1) In call by reference, address of the variable is passed as arguments
2) In call by value, pointer is not used	2) In call by reference, pointer is used.
3) In call by value, any changes made in the formal arguments are not reflected back to the actual arguments	3) In call by reference, any changes made in the formal arguments are reflected back to the actual arguments.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>

void swap(int *,int *);           //Function Declaration

void main()
{
    int x,y;
    clrscr();
    cout<<"Enter x";
    cin>>x;
    cout<<"Enter y";
    cin>>y;
    swap(&x,&y);                   //Function Call
    getch();
}

void swap(int *a,int *b)         //Function Definition
{
    int c;
    c=*a;
    *a=*b;
    *b=c;
    cout<<a;
    cout<<b;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	The arguments which are passed in function definition are known as	Formal
2.	The arguments which are passed in function call are known as.....	Actual
3.	In call by value, any change made in formal arguments are reflected back to actual arguments?? (T/F)	False
4.	In call by Reference, any change made in formal arguments are reflected back to actual arguments?? (T/F)	True
5.	In call by, pointer is not used	Call By Value

Topic: Explain return by reference

Ans:.

- As the functions have different return types, in the same way function can also return the reference.
- In return by reference, the function is called on the left side of the assignment.
- In the below example, $\max(x,y)=100$ is called. It returns reference to the variable holding the maximum value and assigns the value 100 to it. If the value of a is greater than b, 100 is assigned to a otherwise 100 is assigned to b.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>

int &max(int &,int &); //Function Declaration

void main()
{
    int x,y;
    clrscr();
    cout<<"Enter x";
    cin>>x;
    cout<<"Enter y";
    cin>>y;
    max(x,y)=100; //Function Call
    cout<<x;
    cout<<y;
    getch();
}
int &max(int &a,int &b) //Function Definition
{
    if(a>b)
    return a;
    else
    return b;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	Return by reference means?	Function returns the pointer or reference as in same way as it returns the value.
2.	In return by reference, Function is called on left side of	Assignment
3.	In return by reference, value must be initialized?	Yes
4.	$\max(x,y)=100$.What does this indicate?	The value (either x or y) that is greater will be updated to 100 .

Topic: What is Inline Function?

Ans:

- In the normal function call that is when the function is called, the cursor switches over to the function definition and then each and every statement inside the function definition is executed. This increases the function call overhead.
- If 5 times same function is called then same process is repeated for 5 times.
- To overcome this disadvantage or to reduce the function call overhead, inline function is used.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- Inline function is a function that is expanded in a line when the function is called.
- When the call to the inline function is made, the cursor switches over to the function definition but statements are not executed, instead they are copied and the statements are pasted where the function is called.

Syntax:

```
inline returntype functionname(parameters)
{
    .//Function Code
}
```

- Inlining is only a request to the compiler, not a command. Compiler can ignore the request for inlining.
- Compiler may not perform inlining in such circumstances like:
 - 1) if a function contains a loop. (for, while, do-while)
 - 2) If a function contains static variables.
 - 3) If a function is recursive.
 - 4) If a function return type is other than void, and the return statement doesn't exist in function body.
 - 5) If a function contains switch or goto statement.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>

void main()
{
    print();
    print();
    getch();
}
inline void print()           //inline function
{
    cout<<"Hello";
}
```

Sr No.	Question	Answer
1.function is used to reduce function call overhead	Inline
2.	Inline function is only a request to compiler, not a command (T/F)	True
3.	Inline function does not support loop (T/F)	True



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

4.	Inline function does not support switch or goto statement?	True
5.	Inline function should not be recursive? (T/F)	True
6.keyword is used to make the function inline	inline

Topic: Default Arguments:

Ans:

- A default argument is a value provided in a function declaration that is automatically assigned by the compiler if the caller of the function doesn't provide a value for the argument with a default value.
- Default arguments should be always passed in function declaration.
- Default arguments cannot be assigned from middle. It should be assigned from left to right.
- In below example, when add() is called with no arguments, all 3 arguments value is taken as default value. When add(1) is called with 1 argument, y and z value is taken as default value. When add(2,3) is called with 2 arguments, z value is taken as default value. When add(1,2,3) is called with 3 arguments, no value is taken as default value



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>

int add(int x=10, int y=15, int z=20); //Func. Declaration

void main()
{
    cout<<add(); //Function with no arguments
    cout<<add(1); //Function with 1 arguments
    cout<<add(2,3) //Function with 2 arguments
    cout<<add(1,2,3) //Function with 3 arguments
    getch();
}
int add(int x,int y, int z)    //Function Definition
{
    int sum;
    sum=x+y+z;
    return sum;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.arguments are provided by compiler when the variables are not given the value	Default
2.	Default arguments are always passed in function	Declaration
3.	Default arguments should be initialized from	Left to Right
4.	Default arguments cannot be initialized from middle? (T/F)	True

Topic: Explain const arguments

Ans:

- A const argument means the argument which cannot be modified or changed.
- **Const keyword** is used to define the constant argument.
- The keyword const specifies that the value of variable will not change throughout the program.
- If you try to change the value of variable that is defined constant, it will generate an error



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
```

```
Void change (const int j) //Function Definition
```

```
{
```

```
    j=30;
```

```
}
```

```
Void main ()
```

```
{
```

```
    int i=10;
```

```
    clrscr();
```

```
    change(i); //Function call
```

```
    cout<<endl<<I;
```

```
    getch();
```

```
}
```

Sr No.	Question	Answer
1.	Constant argument cannot be changed or modified? (T/F)	True
2.keyword is used to declare constant arguments.	const



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: What is Function Overloading

Ans:

- Multiple functions with same name but different parameters is known as Function Overloading.

Example:

```
#include<iostream.h>
#include<conio.h>

void sum(int a,int b)
{
    int c;
    c=a+b;
    cout<<c;
}
void sum(int a,int b,int c)
{
    int d;
    d=a+b+c;
    cout<<d;
}

void main()
{
    clrscr();
    sum(5,5);
    sum(1,2,3);
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	Multiple functions with same name but different parameters is known as.....	Function overloading
2.	Different parameters include?	Data type and Number of paramters

Topic: Explain C Structure Revisited

Ans:

- In C Language, structure is the collection of variables or members which may or may not have same data type.
- C language does not support OOP, So all the members of structure are by default **public**.

Example:

```
struct student  
{  
    int roll;  
    char name[10];  
};
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- While in c++, the concept of structure is overwritten by class.
- Class is the collection of members or variables which may or may not have same data type.
- But as c++ supports OOP, so we can use public and private access specifier as per our requirement.

Example:

```
class student
{
    public:
    int roll;
    private:
    char name[10];
};
```

- So, in the above example, roll is declared as public so all the functions in the program can access roll variable but name is declared as private so all the functions in the program cannot access name variable.

Structure	Class
1) Structure does not support OOP	1) Class supports OOP
2) Structure does not support public or private access sepcifier	2) Class supports public or private access sepcifier



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	All the members of structure are by default	static
2.	Structure supports OOP? (Y/N)	No
3.	Class supports public or private access specifier? (Y/N)	Yes

Topic: Write a short note on local class.

Ans:

- The class which is declared inside the function is known as local class.

Example showing how to declare local class:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
void print()
{
    class test
    {
        public:
        void mymethod()
        {
            cout<<"Hello";
        }
    };
    test t;
    t.mymethod();
}

void main()
{
    print();
    getch();
}
```

Features of Local Class:

- 1) Local class type name (object) should be declared inside the function only in which local class is declared.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
void print()
{
    class test
    {
        public:
        void mymethod()
        {
            cout<<"Hello";
        }
    };
    test t;
    t.mymethod();
}

void main()
{
    print();
    getch();
}
```

Object declared inside function only

2) The methods of local class should be declared inside local class only.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
void print()
{
    class test
    {
        public:
        void mymethod()
        {
            cout<<"Hello";
        }
    };
    test t;
    t.mymethod();
}

void main()
{
    print();
    getch();
}
```

⇒ mymethod() should be defined inside class test only because class test is local class



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

3) Local class cannot access static variables but can access static methods.

Example:

```
#include<iostream.h>
#include<conio.h>
void print()
{

    class test
    {
        static int a;           Can not access static variable
        public:
        static void mymethod() can access static method
        {
            cout<<"Hello";
        }
    };
    test t;
    t.mymethod();
}
void main()
{
    print();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

4) Local class can access global variables and also can access local class of same function

Example:

```
#include<iostream.h>
#include<conio.h>
int x;
void print()
{
    class test1
    {
        public:
        void mymethod1()
        {
            cout<<"Hello";
        }
    };
    class test2
    {
        test1 t1;
        t1.mymethod1(); //method of local class test1
        public:
        void mymethod2()
        {
            cout<<"Hi";
        }
    };
    test2 t2;
    t2.mymethod2 ();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	The class declared inside the function is known as	Local Class
2.	Local class can accessvariables	Global
3.	Local class cannot access but can access	Static Variables Static Methods
4.	The methods of local class should be declared inside the local class only. (T/F)	True

Topic: Write a short note on nested class.

Ans:

- The class inside another class is known as nested class.
- To create the object of nested class we have to use scope resolution operator.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
class A
{
    public:
    class B Nested Class };
    {
        private:
        int num;
        public:
        void getdata(int n)
        {
            num=n;
        }
    }
    void putdata()
    {
        cout<<num;
    }
};
Scope Resolution operator
void main()
{
    A::B obj;
    obj.getdata(5);
    obj.putdata();
    getch();
}
```

Sr No.	Question	Answer
1.	Class inside another class is known asclass	Nested
2.	Object of nested class is created using operator	:: (Scope Resolution)



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

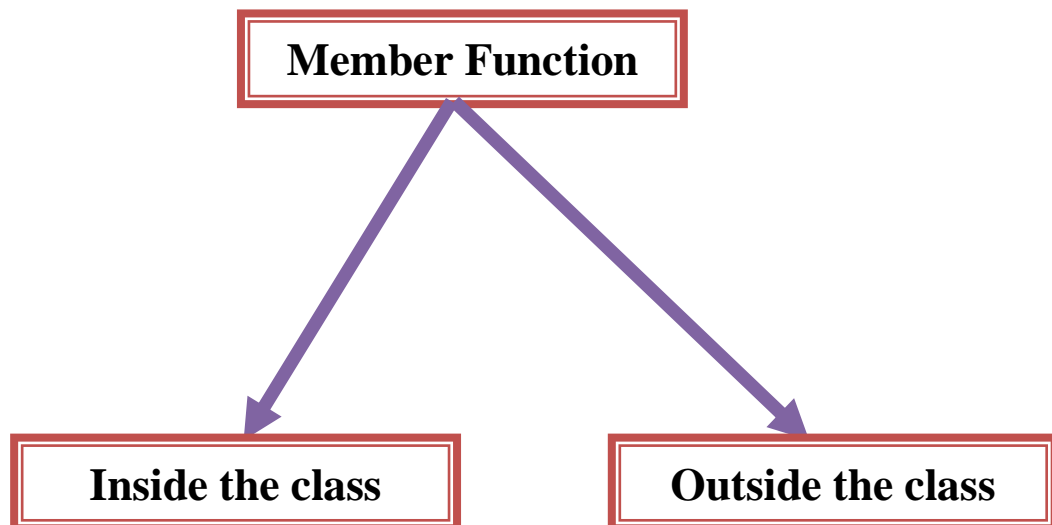
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain member function inside and outside the class.

Ans:

- Member functions are the functions which have their declaration inside the class and works on the data members of the class.
- The definition of member function can be inside or outside the class.



Inside the class:

- If the member function is defined inside the class definition it can be defined directly



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
class square
{
    public:
    void getsquare(int a)
    {
        cout<<"Square is"<<(a*a);
    }
};

void main()
{
    square s;
    s.getsquare(5);
    getch();
}
```

Outside the class:

- But if we want to define the member function outside the class definition then we must declare the function inside class definition and then define it outside.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- When we define the function outside the class, we have to use scope resolution operator.

Syntax:

```
class classname
{
    Function declaration
    returntype memberfunction(parameters);
};
returntype classname::memberfunction(parameters)
{
    //body;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
class square
{
    public:
    void getsquare(int); //Function delcaration
};

void square::getsquare(int a)
{
    cout<<"Square is"<< (a*a);
}

void main()
{
    square s;
    s.getsquare(5);
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	Member functions have their declarations inside the	Class
2.	The function whose definition is inside the class is known as	Member Function inside the class
3.	The function whose definition is outside the class is known as.....	Member Function outside the class
4.	When the member Function is declared outside the class we have to useoperator	:: (Scope Resolution Operator)

Topic: Explain nesting of member functions

Ans:

- A member function of a class can be called only by an object of that class using a dot operator. However, there is an exception to this.
- A member function can be called by using its name inside another member function of the same class. This is known as nesting of member functions.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class nest
{
    public:
    void square(int a)
    {
        cout<<"Square is"<<(a*a);
    }
    void cube(int a)
    {
        cout<<"Cube is"<<(a*a*a);
    }
    void display()
    {
        square(5);
        square(5);
    }
};

void main()
{
    nest n1;
    n1.display();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	Member function inside another member function is known as	Nesting of Function

Topic: Write a short note on private member function.

Ans:

- A function declared inside the class's private section is known as private member function.
- A private member function is accessible only through the public member function.

Example:

```
File Edit Search Run Compile Debug Project Options Window Help
NONAME05.CPP
//Program to demonstrate private member function_
class test
{
    private:
    void print()
    {
        cout<<"Hello";
    }
    public:
    void display()
    {
        print();
    }
};

void main()
{
    test t1;
    t1.display();
    getch();
}
```

5:49

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	A function declared inside the private section is known as	Private Member Function
2.	A private member function is accessible only throughmember functions.	public

Topic: Write a short note on array within class

Ans:

- Arrays can be declared as members of the class.
- The arrays can be declared as public, private or protected members of the class.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
#define size 5
class student
{
    int roll;
    int marks[size];
public:
    void getdata();
    void total();
};
void student::getdata()
{
    cout<<"Enter rollno";
    cin>>roll;
    for(int i=0;i<size;i++)
    {
        cout<<"Enter marks";
        cin>>marks[i];
    }
}
void student::total()
{
    int t;
    for(int i=0;i<size;i++)
    {
        t=t+marks[i];
    }
    cout<<"Total marks is"<<t;
}
}
```

```
void main()
{
    student s;
    s.getdata();
    s.total();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1)	Can class contain array? (Y/N)	Yes

Topic: Write a short note on Memory allocation of objects

Ans:

- The object is the variable which is defined over data type class. A class contains both member variables and member functions.
- The memory allocated to the object is calculated using sizeof operator
- The memory required by object is sum of memory required by its data members (variables)



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class nest
{
    int roll;
    char name[10];
    public:
    void getdata()
    {
        cout<<"Enter roll";
        cin>>roll;
        cout<<"Enter name";
        cin>>name;
    }
    void display()
    {
        cout<<"Rollno is"<<roll;
        cout<<"Name is"<<name;
    }
};
void main()
{
    nest n1;
    n1.getdata();
    n1.display();
    cout<<"Memory of object n1 is"<<sizeof(n1);
    cout<<"Memory of object n2 is"<<sizeof(n2);
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	Memory is allocated to objects using	sizeof()
2.	The memory required by object is sum of memory required by its data members (variables)? (T/F)	True

Topic: Write a short note on Static Data Members

Ans:

- Static data members are the data members which are accessible to all of the objects of the class, such as flag variable which states whether some specific initialization has occurred.

Syntax:

```
Class classname  
{  
    ....  
    static datatype datamember;  
};
```

Note: The static data member is initialized out of the class.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax:

Datatype classname :: datamember=Initialization;

Features of static data members:

- 1) When first object of class is created, it is initialized to 0 by default.
- 2) Only one copy of static data member is created for entire class and is shared by all objects of that class.
- 3) It is visible only within the class, but its lifetime is the entire program

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
#include<isotream.h>
#include<conio.h>

class test
{
    static int count;
    public:
    void setdata()
    {
        count++;
        cout<<"Counter value is"<<count;
    }
};
int test::count=0;
void main()
{
    test t1,t2;
    t1.setdata();
    t2.setdata();
    getch();
}
```

1



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	Static data members are accessible to all the objects? (Y/N)	Yes
2.	Only ...copy of static variables is created.	1
3.	Static variables is visible onlybut its lifetime is	inside the class throughout the program
4.	Static data members are initialized outside the class. (T/F)	True

Topic: Write a short note on Static Member Function

Ans:

- Same as static member variable, class can also have static member function.

Features of Static Member Function:

- 1) Only 1 copy of static member function exists for all instance of a class.
- 2) A static member function can access only static data member
- 3) A static member function can be called directly using class name. Object of the class is not required.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<isotream.h>
#include<conio.h>

class test
{
    static int count;
public:
    void setdata()
    {
        count++;
    }
    static void display()
    {
        cout<<"Counter is"<<count;
    }
};
int test::count=0;
void main()
{
    test t1,t2;
    t1.setdata();
    test::display();
    t2.setdata();
    test::display();
    getch();
}
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	A static member function can access only	static data member
2.	A static member function can be called directly using	classname

Topic: Write a short note on Array of Objects

Ans:

- Array of objects means creating the array of the class data type.
- The array of class type contains the object of the class as its individual elements
- Array of class type is also known as array of objects.

Syntax:

Classname classobjectname[arraysize];

Example:

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
#include<isotream.h>
#include<conio.h>
```

```
class test
{
    int x;
    public:
    void setdata()
    {
        x=i;
    }
    void getdata()
    {
        cout<<x;
    }
};
```

```
void main()
{
    test t1[4];
    int l;
    for(i=0;i<4;i++)
    t1[i].setdata();
    for(i=0;i<4;i++)
    t1[i].getdata();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1)	Array of object means?	creating array of class data type.

Topic: Write a short note on Friend Function

Ans:

- A friend function can access protected and private data of a class.
- By using keyword friend, compiler knows the given function is a friend function.
- For accessing the data, the declaration of a friend function should inside the body of a class starting with the keyword friend.

Syntax:

```
class classname
{
    friend datatype functionname(arguments);
};
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- In the above declaration, the friend function is preceded by the keyword friend. The function can be defined anywhere in the program like a normal C++ function. The function definition does not use either the keyword **friend** or **scope resolution operator**.

Characteristics of a Friend function:

- The function is not in the scope of the class to which it has been declared as a friend.
- It cannot be called using the object as it is not in the scope of that class.
- It can be invoked like a normal function without using the object.
- It cannot access the member names directly and has to use an object name and dot membership operator with the member name.
- It can be declared either in the private or the public part.



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
class test
{
    private:
    int a,b;
    public:
    void getdata()
    {
        cout<<"Enter a";
        cin>>a;
        cout<<"Enter b";
        cin>>b;
    }
    friend void add();
};
void add()
{
    int temp;
    test t;
    t.getdata();
    temp=s.a+s.b;
    cout<<"Sum is"<<temp;
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1)	A friend function can accessandmembers of the class	Private and Protected
2)	A friend function is declared usingkeyword	friend
3)	A friend function definition can not contain scope resolution operator or friend keyword (T/F)	True
4)	A friend function does not require the object when called. (T/F)	True



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on passing the objects as function arguments.

Ans:

- As normal variables are passed in the function, same way objects are also passed as function.



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
```

```
class Demo
```

```
{
```

```
    private:
```

```
    int a;
```

```
    public:
```

```
        void set(int x)
```

```
        {
```

```
            a=x;
```

```
        }
```

```
    }
```

```
    void sum(Demo d1,Demo d2)
```

```
    {
```

```
        a=d1.a+d2.a;
```

```
    }
```

```
    void display()
```

```
    {
```

```
        cout<<a;
```

```
    }
```

```
};
```

```
void main()
```

```
{
```

```
    Demo d1;
```

```
    Demo d2;
```

```
    Demo d3;
```

```
    d1.set(10);
```

```
    d2.set(11);
```

```
    d3.sum(d1,d2);
```

```
    d3.display();
```

```
    getch();
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on returning object.

Ans:

- As, function returns the value similarly a function can return object.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
```

```
class Demo
{
```

```
    private:
```

```
    int a;
```

```
    public:
```

```
        void set(int x)
```

```
        {
```

```
            a=x;
```

```
        }
```

```
    }
```

```
        Demo sum(Demo d1,Demo d2)
```

```
        {
```

```
            Demo d3;
```

```
            d3.a=d1.a+d2.a;
```

```
            return d3;
```

```
        }
```

```
        void display()
```

```
        {
```

```
            cout<<a;
```

```
        }
```

```
};
```

```
void main()
```

```
{
```

```
    Demo d1,d2,d3;
```

```
    d1.set(10);
```

```
    d2.set(11);
```

```
    d3=d3.sum(d1,d2);
```

```
    d3.display();
```

```
    getch();
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

SUBJECT: C++

CHAPTER-3:

Sr No.	Topic
1.	Constructor
2.	Types of constructor
3.	Constructor Overloading
4.	Constructor with default arguments
5.	Dynamic Initialization of Objects
6.	Dynamic Constructor
7.	Destructor
8.	Operator Overloading 1) What is operator overloading? 2) Features/Rules of operator overloading. 3) Overloading Binary Operator 4) Overloading Unary Operator 5) Overloading unary operator using friend function
9.	Type Conversion
10.	MIL



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: What is Constructor?

Ans:

- Constructor is a special member function that enables an object to be initialized when it is created
- A constructor has same name as class name without return type not even void.
- The constructor can be defined inside or outside the class.
- The constructor must be defined inside the public part.

Syntax to define constructor inside the class:

```
class classname
{
    private:
    data member

    public:
    classname()    //Constructor
    {
        ....
        ...
    }
};
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax to define constructor outside the class:

```
class classname
{
    private:
        data members;
    public:
        classname();
};
classname::classname()//Constructor
{
    //Body of constructor
}
```

Characteristics of Constructor:

1. It has same name as that of class name to which it belongs
2. It is executed automatically whenever the object of the class is created
3. It does not have any return type not even void
4. It is normally used to initialize the data member if class
5. The address of constructor cannot be referred.
6. It cannot be inherited
7. It makes use of new and delete operator



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h.
class test
{
    int x,y;
calling
public:
test() //Constructor
{
    x=10;
    y=10;
}
void output()
{
    cout<<"x=x;
    cout<<"y=y;
}
}

void main()
{
    test t1; //Constructor
    t1.output();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

In the above example, the constructor used is default constructor

M.C.Q		
Sr No.	Question	Answer
1.	Constructor have same name as class name? (T/F)	True
2.is used to initialize the values.	Constructor
3.	Constructor does not have any return type not even void. (T/F)	True
4.	Constructor is executed automatically when	Object of the class is created
5.	Can Constructor be inherited?	No

Topic: Types of Constructor

Ans:

- Following are the types of constructor:

- 1) Default Constructor
- 2) Parameterized Constructor
- 3) Copy Constructor

1) Default Constructor:

- The constructor with no arguments or 0 arguments is known as default constructor.
- If no constructor is defined for the class, then default constructors is called.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
class test
{
    int x,y;
public:
    test() //Constructor
    {
        x=10;
        y=10;
    }
    void output()
    {
        cout<<"x=x;
        cout<<"y=y;
    }
};

void main()
{
    test t1; //Constructor calling
    t1.output();
    getch();
}
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrपाल Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

2) Parameterized Constructor

- The constructor with arguments is known as parameterized constructor.
- When parameterized constructor is used, proper arguments or parameters must be provided.

Example:

```
#include<iostream.h>
#include<conio.h>
class test
{
    int x,y;
    public:
    test(int x1, int y1) //Constructor
    {
        x=x1;
        y=y1;
    }
    void output()
    {
        cout<<"x=x;
        cout<<"y=y;
    }
};

void main()
{
    test t1(5,5); //Constructor calling
    t1.output();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- A Constructor can have argument of any data type.
- When the constructor has object of its own class, is known as copy constructor.
- The object of the own class must be passed as reference parameter.

Example:

```
test(test &obj)
{
}
}
```

- In the above example, the constructor is having a reference to the instance of its own class as an argument and is known as copy constructor.
- A compiler copies all the members of the user defined source object to the destination object in the assignment statement.
- A copy constructor can be called like this way:

```
test t1(5);
test t2(t1) or test t2=t1
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
class test
{
    int x,y;
    public:
    test(int x1, int y1)
    {
        x=x1;
        y=y1;
    }
    test (test &t)
    {
        a=t.a;
        b=t.b;
    }
    void output()
    {
        cout<<"x=x;
        cout<<"y=y;
    }
}

void main()
{
    test t1(5,5);
    t1.display();
    test t2(t1);
    t2.display();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.constructor is called when no constructor is called	default
2.	The constructor having parameters is known as	Parameterized constructor
3.	Default constructor is also known as	zero argument constructor
4.	Inconstructor, reference of object is passed	copy
5.	Inconstructor, the data of 1 object is copied into other object	copy

Topic: Write a short note on constructor overloading

Ans:

- Constructor with same name but different parameters is known as constructor overloading.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
#include<iostream.h>
#include<conio.h>
```

```
class test
{
    int a,b;
    public:
    test()
    {
        a=0;
        b=0;
    };
    test(int a1,int b1)
    {
        a=a1;
        b=b1;
    }
};
```

```
test(test &t)
{
    a=t.a;
    b=t.b;
}
void display()
{
    cout<<a<<b;
}
};
```

```
void main()
{
    test t1;//default
    t1.display();
    test t2(5,6);
    t2.display();
    test t3(t2);
    t3.display();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	Can a program have multiple constructors?	Yes
2.	Constructor with same name but different parameters is known as	Constructor overloading

Topic: Write a short note on constructor with default arguments.

Ans:

- Default arguments are the arguments that are passed when the user does not give any value.
- Constructor can also be used with default arguments.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iostream.h>
#include<conio.h>
class test
{
    int a,b;
    public:
    test(int a1,int b1=10)
    {
        a=a1;
        b=b1;
    };
    void display()
    {
        cout<<a<<endl<<b;
    }
};

void main()
{
    test t1(5);
    t1.display();
    getch();
}
```

- In the above example, when constructor is called, only 5 is passed and that value is copied into a1 variable, but the value of b is not passed so in this case, compiler will default value of b1.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1)	Can we pass the default arguments to constructor?	Yes

Topic: Write a short note on dynamic initialization of objects.

Ans:

- A class object can also be initialize at run time. It is known as dynamic initialization of object.
- One advantage of dynamic initialization is that with the multiple constructors various initialization techniques can be provided.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class test
{
    int x;
    public:

    test()
    {

    }
    test (int a)
    {
        x=a;
    }
    int add()
    {
        return (x+2);
    }
    void display()
    {
        cout<<x;
    }
};

void main()
{
    int a;
    cout<<"Enter a";
    cin>>a;
    test t1(a);
    t1.display();
    test t2(t1.add());
    t2.display();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- When allocation of memory is done dynamically using dynamic memory allocator new in a constructor, it is known as dynamic constructor.

```
class test
{
    char *p;
public:
    test()
    {
        p=new char[6];
        p="hello";
    }
    void display()
    {
        cout<<p;
    }
};

void main()
{
    test t1;
    t1.display();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	The constructor which allocates memory dynamically is known as	dynamic Constructor
2.keyword is used to allocate the memory dynamically	new

Topic: Write a short note on destructor

Ans:

- The constructor is used to initialize the data member and allocate memory for object at time of creation.
- When object is no longer needed it can be destroyed. A class can have special member function that is executed when the object is destroyed
- Destructor is a special member function that has same name as class name but is indicated with ~ symbol. And it is used to destroy object
- Destructor is automatically called when object is destroyed.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax:

```
class classname
```

```
{
```

```
    public:
```

```
    ~ classname()
```

```
    {
```

```
    }
```

```
};
```

} DESTRUCTOR

Features of Destructor:

- 1) It has same name as class name
- 2) It is preceded with symbol ~
- 3) It is called automatically when object is destroyed
- 4) It is also declared in public section
- 5) Destructor cannot have any arguments or parameters
- 6) There can be only 1 destructor in each class.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
class test
{
    int x;
public:
    test()
    {
        x=20;
    }
    ~test()
    {
        cout<<"Destructor called";
    }
    void display()
    {
        cout<<x;
    }
};

void main()
{
    test t;
    t.display();
    cout<<"Terminating";
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Difference between constructor and destructor.

Constructor	Destructor
<ul style="list-style-type: none">A class can have multiple constructors.	1) A class can have only 1 destructor.
2) Constructors can be overloaded	2) Destructor cannot be overloaded.
3) Constructors are not preceded by ~	3) Destructor is preceded by ~
4) Constructor is called when object of class is created	4) Destructor is called when object of the class is destroyed.
5) Constructor can have arguments or parameters	5) Destructor can not have arguments or parameters

M.C.Q		
Sr No.	Question	Answer
1.	Can Destructor overload? (Y/N)	No
2.	Destructor is preceded bysymbol	~
3.	Destructor is called when.....	object is destroyed



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on operator overloading

Ans:

- Operator overloading is a mechanism, which provides a special meaning to an operator.
- The operator overloading is one of the method of polymorphism. Polymorphism means many forms and performs many actions with a single operator.

Syntax to define operator overloading:

```
returntype operator op(argument list)
{
    //Body
}
```

- In the above syntax, returntype is the type of value returned by the specified operation, Operator is the keyword and op is the operator which want to overload.

Rules of operator overloading:

- 1) Overloaded operators must perform operation similar to those defined for basic data types that is + operator can be overloaded to perform addition. It cannot be redefined to subtract the values.
- 2) Only existing operators can be overloaded. New operator cannot be created
- 3) Atleast one operand should be of user defined data type such



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

as object of class

- 4) Operator function can be either member function or friend function.
- 5) In case of member function, it will take no arguments for unary operator and 1 argument for binary operator
- 6) In case of friend function, it will take 1 argument for unary operator and 2 arguments for binary operator.
- 7) Following operators cannot be overloaded

.	Member accessing operator
.*	Pointer to member operator
::	Scope Resolution operator
?	Conditional Operator
sizeof ()	sizeof operator

- 8) Some of the operators cannot be overloaded by friend function but member function can overload it. They are:

=	Assignment operator
()	Function all operator
[]	Subscripting operator
→	Class member access operator



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example: Binary Operator overloading

```
class test
{
    int x,y;
    public:
    void input()
    {
        cout<<"Enter x and y";
        cin>>x>>y;
    }
    void output()
    {
        cout<<x<<y;
    }
    test operator + (test c)
    {
        test t;
        t.x=x+c.x;
        t.y=y+c.y;
        return (t);
    }
}

void main()
{
    test t1,t2,t3;
    t1.input();
    t2.input();
    t3=t1+t2;
    t3.output();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	Operator overloading is the concept of	Polymorphism
2.	Operator overloading useskeyword	Operator
3.	We can overloadandoperators	Unary and Binary
4.	Only inbuilt operators can be overloaded (T/F)	True
5.	Even though we overload the inbuilt operators, we cannot change the meaning of the operator. (T/F)	True

Example: Unary Operator overloading



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class test
{
    int x,y;
    public:
    void input()
    {
        cout<<"Enter x and y";
        cin>>x>>y;
    }
    void output()
    {
        cout<<x<<y;
    }
    operator - ( )
    {
        x=-x;
        y=-y;
    }
};

void main()
{
    test t1;
    t1.input();
    t1.output();
    -t1;
    t1.output();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	When you are overloading unary operator using simple function, it should containarguments	0
2.	To overload unary operators it should be prefixed bykeyword	operator



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on overloading unary operator using friend function.

Ans:

- Friend function can be used for both unary and binary operators.
- As friend function is going to overload unary operator it will take only 1 argument.

Syntax:

```
friend returntype operator operatorname(arguments)
{
    //Body of function
}
```

- According to above syntax, the friend function must be prefixed with keyword friend inside the class specification. The body of friend function can be either inside class or outside the class. When the definition of friend function is outside the class then it is defined as normal function and not prefixed with friend keyword
- The arguments of friend functions are generally objects of friend function.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class test
{
    int x,y;
public:
    void input()
    {
        cout<<"Enter x and y";
        cin>>x>>y;
    }
    void output()
    {
        cout<<x<<y;
    }
    friend void operator –(test &t);
};

void operator –(test &t)
{
    t1.x=-t1.x;
    t1.y=-t1.y
}

void main()
{
    test t2;
    t2.input();
    t2.output();
    -t2
    t2.output();
    getch();
}
```

In the above example, t2 object is a reference of object t



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on overloading Binary operator using friend function.

Ans:

- Friend function can be used for both unary and binary operators.
- As friend function is going to overload binary operator so it will take 2 arguments or parameters.

Syntax:

```
friend returntype operator operatorname(argument1, argument 2)
{
    //Body of function
}
```

- According to above syntax, the friend function must be prefixed with keyword friend inside the class specification. The body of friend function can be either inside class or outside the class. When the definition of friend function is outside the class then it is defined as normal function and not prefixed with friend keyword

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class test
{
    int x,y;
    public:
    void input()
    {
        cout<<"Enter x and y";
        cin>>x>>y;
    }
    void output()
    {
        cout<<x<<y;
    }
    friend void operator +(test t1,test t2);
};

void operator +(test t1,test t2)
{
    test t3;
    t3.x=t1.x+t2.x
    t3.y=t1.y+t2.y;
    return (t3);
}
```

```
void main()
{
    test t1,t2,t3
    t1.input();
    t2.input();
    t3=t1+t2;
    t3.output();
    getch();
}
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	Can we do operator overloading using friend function?	True
2.	Operator overloading when done with friend function, keywordis used to declare operator overloading function as friend?	friend
3.	When doing operator overloading using friend function, unary operator overloading should havearguments and binary operator overloading should havearguments.	One Two

Topic: Write a short note on Type Conversion

Ans:

- Type conversion means converting one data type into another data type when in a single statement, we have variables with different data types.
- There are two types of type Conversion:
 - 1) Implicit Conversion
 - 2) Explicit Conversion

1) Implicit Conversion:

- Implicit conversion is achieved by assignment operator (=). The assignment operator assigns the content of a variable, result of expression, or constant to another variable.
- When both the variables are of different data type then
int ;



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
float b=5.213;  
a=b;
```

- Where a is type of int and b is of type float, the compiler calls a special routine to convert the value of b to an integer format automatically.

Note: Implicit Conversion is involved only with built-in data types.

2) Explicit Conversion:

- Explicit Conversion is used with user defined data types.
- When in single statement, there are variables with different data types in which one of the data type is user defined data type then explicitly type conversion should be done because compiler does not know anything about user define data type.
- There are three situations in which need of explicit conversion are used.
 - Conversion from basic data type to class type
 - Conversion from class type to basic data type
 - Conversion from one class type to another class type.

**** Conversion from basic data type to class type:**

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- To convert data from basic type to user define type, the conversion function should be defined in class in the form of constructor.
- The constructor can take a single argument whose type is to be converted.

Syntax:

```
Constructor (basic type)
{
    //code for conversion
}
```

Where constructor specifies the same name as class in which it belongs. It has only one argument of basic type.

Example:

```
class test
{
    int x,y,z;
    public:
    test(int a)
    {
        x=a;
        y=a;
        z=a;
    }
    void display()
    {
        cout<<x<<endl<<y<<endl<<z;
    }
};

void main()
{
    test t1(5);
    t1.display();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

basic data type.

- In case of converting user define data type to basic data type, c++ allows you to define operator function.

Syntax:

```
Operator Basic Type ()  
{  
    //code for conversion  
}
```

In the above syntax, operator is the keyword. Basic data type means inbuilt data types.

Note:

- 1) The conversion operator function should not specify a return type.
- 2) The conversion function must not have any arguments
- 3) The conversion function should be declared inside the class only.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class test
{
    int x;
    float y;
public:
    void input()
    {
        cout<<"Enter x,y";
        cin>>x>>y;
    }

    operator float ()
    {
        return (x*y);
    }
};

void main()
{
    test t1;
    float f;
    t1.input();
    f=t1;
    cout<<f;
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

** Conversion from one class to another class type

- The c++ compiler does not support data conversion between objects to user defined classes
- The conversion can be achieved by 2 ways:
 - 1) Constructor
 - 2) Conversion Function

Example: In below example, product is source class and item is destination class.

```
class product
{
    public:
    int m,n;
    void setdata(int x,int y)
    {
        m=x;
        n=y;
    }
};

class item
{
    int a,b;
    public:
    void show()
    {
        cout<<a<<b;
    }
    item()
    {
    }
}

item (product p)
{
    a=p.m;
    b=p.n;
}

void main()
{
    item i1;
    product p1;
    p1.setdata(3,4);
    i1=p1;
    i1.show();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

<u>Conversion from basic type to user type</u>	<u>Conversion from user type to basic type</u>	<u>Conversion from user define type to another user define type</u>
In this, inbuilt data type is converted to user define type.	In this, user define type is converted to inbuilt type.	In this, user define type is converted to another user define type.
In this type of conversion, constructor is used	In this type of conversion, operator function is used	In this type of conversion, constructor and operator function both can be used.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q

Sr No.	Question	Answer
1.	Converting one data type into another data type is known as	Type Conversion
2.	There aretypes of type conversion	2
3.conversion is done automatically usingoperator	Implicit Assingment (=)
4.	In explicit conversion, when converting basic type to user define type.....is used.	Constructor
5.	In explicit conversion, when converting basic type to user define type.....is used.	Operator Function
6.	In the below statement,object is the source object andobject is the destination object $i1=p1$	p1 : Source i1: Destination
7.	Conversion from one user defines type to another user define type can be achieved through and	Constructor Operator Function



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on MIL

Ans:

- MIL stands for Member Initializer List
- It is used to initialize the data members.
- The Initializer list does not end in a semicolon.

Syntax:

```
Constructorname(datatype value1, datatype value2): datamember(value1),datamember  
(value2)  
{  
    ....  
}
```

Example:

```
class Base  
{  
    private:  
    int value;  
    public:  
    Base(int value):value(value)  
    {  
        cout<<"Value is"<<a;  
    }  
};  
void main()  
{  
    Base b1(10);  
    getch();  
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Sr No.	Question	Answer
1.	MIL stands for	Member Initializer List
2.	MIL does not contain semicolon at the end.(T/F)	True
3.	MIL is represented usingsymbol	: (Colon)

Program2:

```
Class point
{
    private:
    int x;
    int y;
    public:
    point(int i=0,int j=0):x(i),y(j)
    {

    }
    void output()
    {
        cout<<x<<y;
    }
};
void main()
{
    point t1(10,15);
    t1.output();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

1) For initialization of non-static const data members.

- Constant data members are initialized using_INITIALIZER list because no separate memory is allocated to constant data members

```
class test
{
    const int t;
public:
    test(int i=0):t(i)
    {

    }
    void show()
    {
        cout<<t;
    }
};

void main()
{
    test t1(10);
    t1.show();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

2) For initialization of reference variables.

- Reference variables must be initialized using_INITIALIZER list.

```
class test
{
    int &t;
    public:
    test(int &t):t(t)
    {

    }
    void show()
    {
        cout<<t;
    }
};

void main()
{
    test t1(10);
    t1.show();
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

3) When no Base class default constructor is present

- In inheritance base class constructor is called followed by child class constructor.

```
class test
{
    public:
    test(int x)
    {
        cout<<x;
    }
};

class B: public test
{
    public:
    B():test(10)
    {
        cout<<"Constructor of B class called";
    }
};

void main()
{
    B b1;
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapal Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

4) When data member and parameter have same name

```
class test
{
    int value;
    public:
    test(int value):value(value)
    {
        cout<<value;
    }
};

void main()
{
    test t1(10);
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

SUBJECT: C++

CHAPTER-4:

Sr No.	Topic
1.	What is Inheritance
2.	What is Derived Class?
3.	Types of Inheritance
4.	Virtual Base Class in c++
5.	Abstract Class
6.	Constructor in derived class
7.	Constructor and destructor in inheritance
8.	Containership
9.	Pointer to objects
10.	Virtual and Pure Virtual Functions
11.	Default Arguments to Virtual Function
12.	RTTI



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

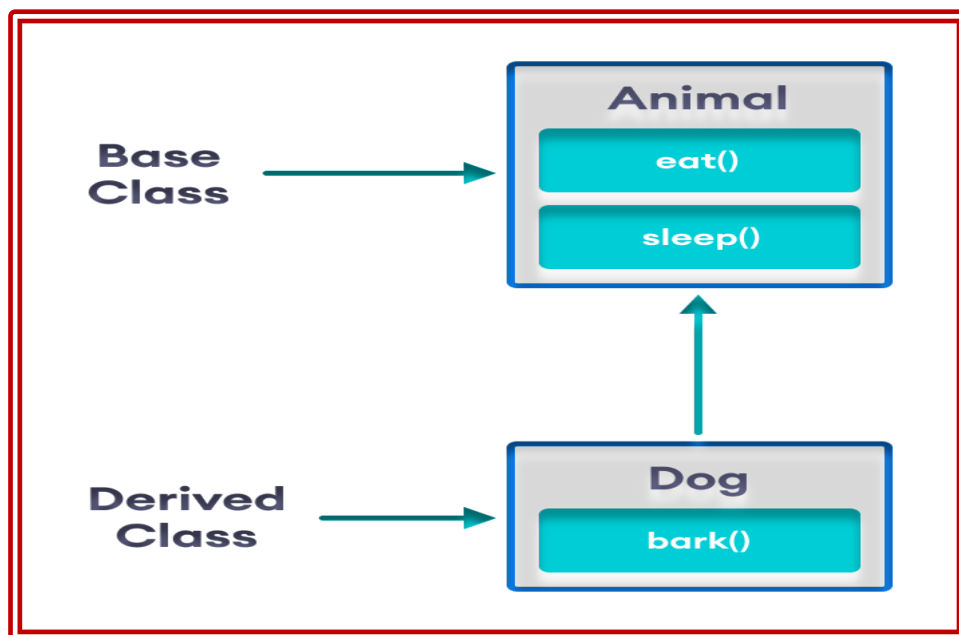
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: What is Inheritance?

Ans:

- It is the process of creating new classes, called derived classes from existing classes, which are often known as base class.
- The derived class inherits all the capabilities of base class and it can also add new features of its own.



- In the above diagram, Animal is one base class which have two methods eat () and sleep ().
- Dog is the derived class which is inherited from Animal class. So it will be able to access eat () and sleep () methods of Animal class as well as it will access its own method bark ().



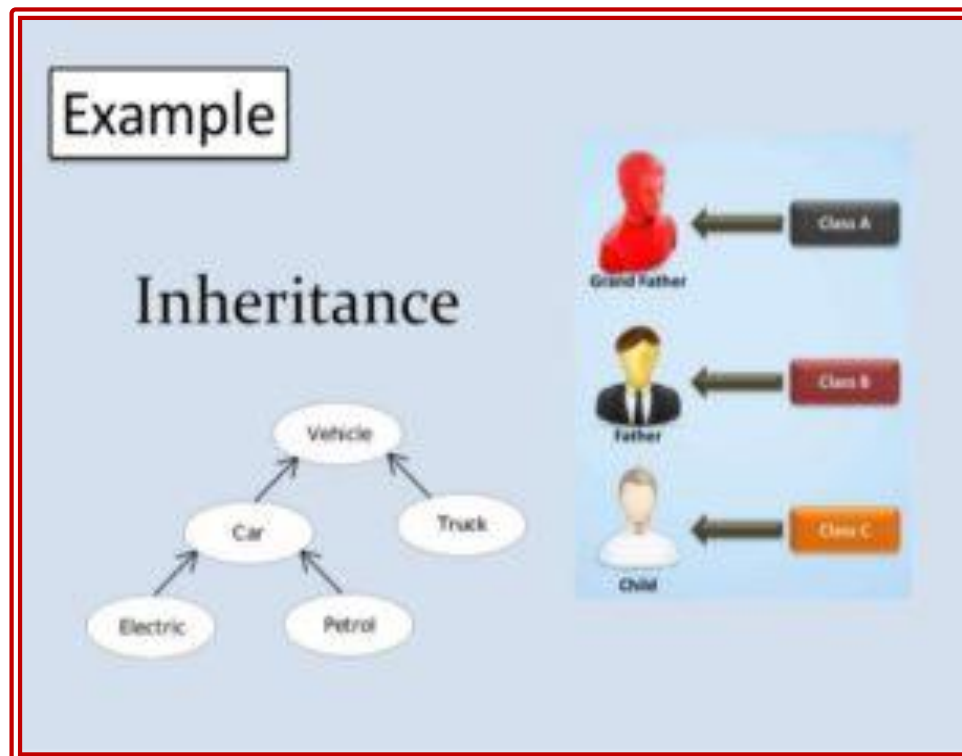
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

NOTE:

- ★ Inheritance is also known as Generalization/Specialization.
- ★ Super class/Base class is known as Generalization and subclass is known as Specialization.



Topic: What is Derived Class?

Ans:

- Derived class extends its features by inheriting properties of base class and also add new features of its own.

Syntax:

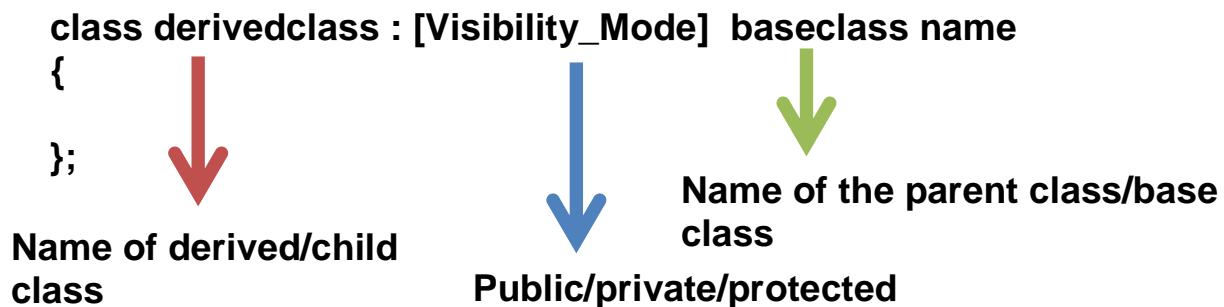
“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645



NOTE:

★ When the classes are inherited publically or privately, the private members of base class are not inherited and therefore it will never become member of its derived class.

Topic: Explain Modes of Inheritance.

Ans:

■ There are 3 modes of inheritance:

- 1) **Public mode:** If we derive a sub class from a public base class. Then the public member of the base class will become public in the derived class and protected members of the base class will become protected in derived class.
- 2) **Protected mode:** If we derive a sub class from a Protected base class. Then both public member and protected members of the base class will become protected in derived class.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

3) **Private mode:** If we derive a sub class from a Private base class. Then both public member and protected members of the base class will become Private in derived class.

Note:

■ The private members in the base class cannot be directly accessed in the derived class, while protected members can be directly accessed. For example, Classes B, C and D all contain the variables x, y and z in below example. It is just question of access.

Base class member access specifier	Type of Inheritance		
	Public	Protected	Private
Public	Public	Protected	Private
Protected	Protected	Protected	Private
Private	Not accessible (Hidden)	Not accessible (Hidden)	Not accessible (Hidden)



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	There aretypes of modes in inheritance	3
2.	The private members of base class cannot be accessed directly (T/F)	True
3.	If we derive a sub class from a Private base class then both public member and protected members of the base class will become in derived class.	Private
4.	If we derive a subclass from protected base class then both public members and protected members of base class will become in derived class	Object of the class is created
5.	If we derive a sub class from a public base class, then the public member of the base class will become in the derived class and protected members of the base class will become in derived class.	Public Protected



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

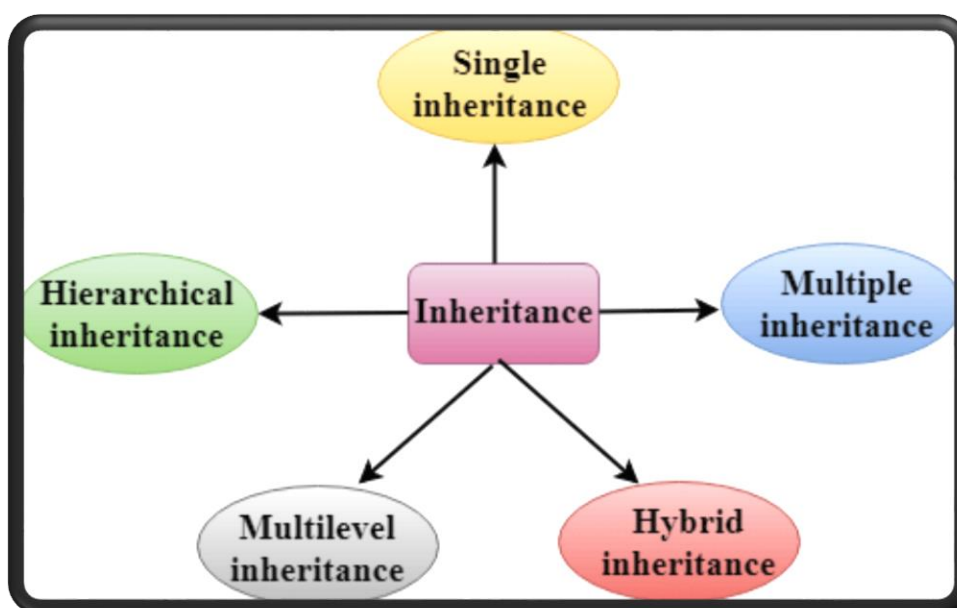
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain types of inheritance.

Ans:

- Inheritance is the transitive process, where the derived class inherits some or all the features of the base class depending on the visibility made.
- Following are the types of inheritance:
 - 1) Single Inheritance
 - 2) Multiple Inheritance
 - 3) Multilevel Inheritance
 - 4) Hierarchical Inheritance
 - 5) Hybrid Inheritance
 - 6) Multiple Inheritance





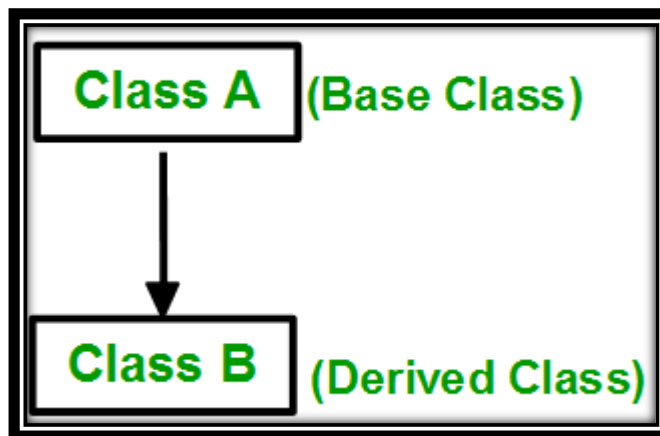
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

1) Single Inheritance:

- ✚ In single inheritance, a class is allowed to inherit from only one class. i.e. one sub class is inherited by one base class only.



Example:

```
class vehicle
{
    public:
    vehicle()
    {
        cout<<"This is Vehicle";
    }
};
class car:public vehicle
{
};
void main()
{
    car c1;
    getch();
}
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

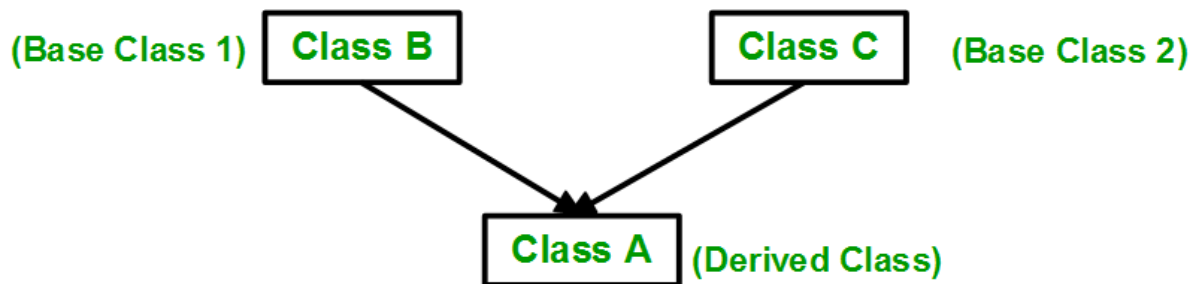
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

}

2) Multiple Inheritance:

- Multiple Inheritance is a feature of C++ where a class can inherit from more than one classes. i.e. one **sub class** is inherited from more than one **base classes**.



Example:

```
class vehicle //Base Class1
{
    public:
    vehicle()
    {
        cout<<"This is Vehicle";
    }
};
class Fourwheeler //Base Class2
{
    public:
    Fourwheeler()
    {
        cout<<"This is fourwheeler";
    }
};
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

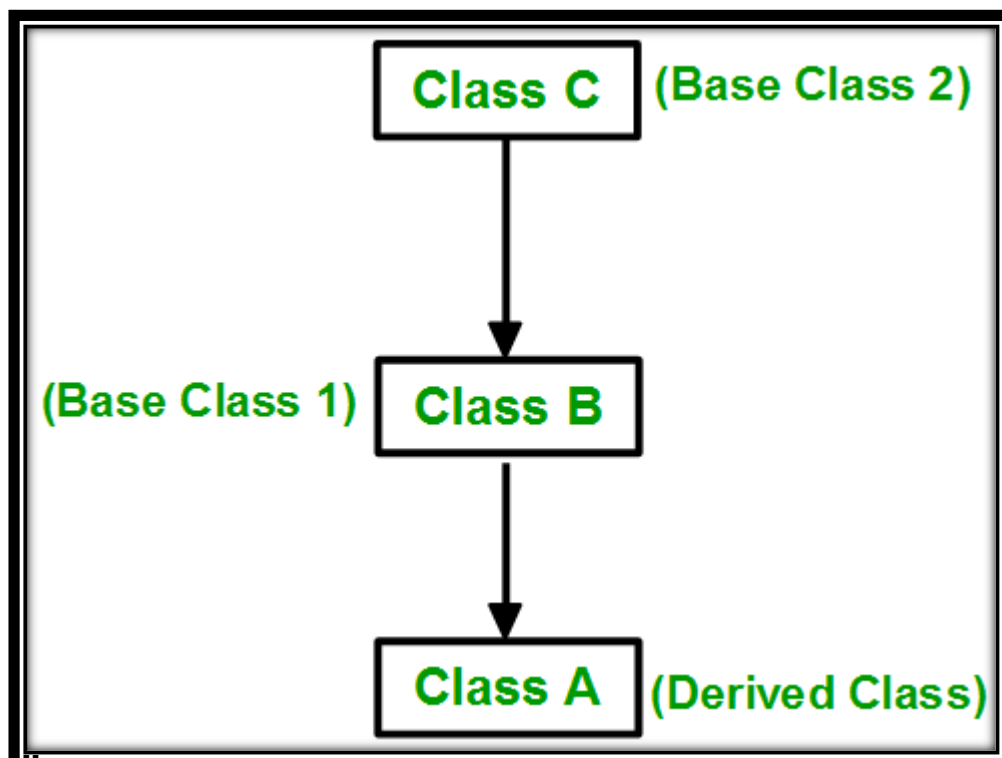
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
};  
class car:public vehicle,public FourWheeler  
{  
  
};  
void main()  
{  
    car c1;  
    getch();  
}
```

3) Multilevel Inheritance:

- ✦ In this type of inheritance, a derived class is created from another derived class.





SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
class vehicle
{
    public:
    vehicle()
    {
        cout<<"This is Vehicle";
    }
};
class Fourwheeler :public vehicle
{
    public:
    Fourwheeler()
    {
        cout<<"This is fourwheeler";
    }
};
class car:public FourWheeler
{
    public:
    car()
    {
        cout<<"Car has 4 wheels";
    }
};
void main()
{
    car c1;
    getch();
}
```

“C++”



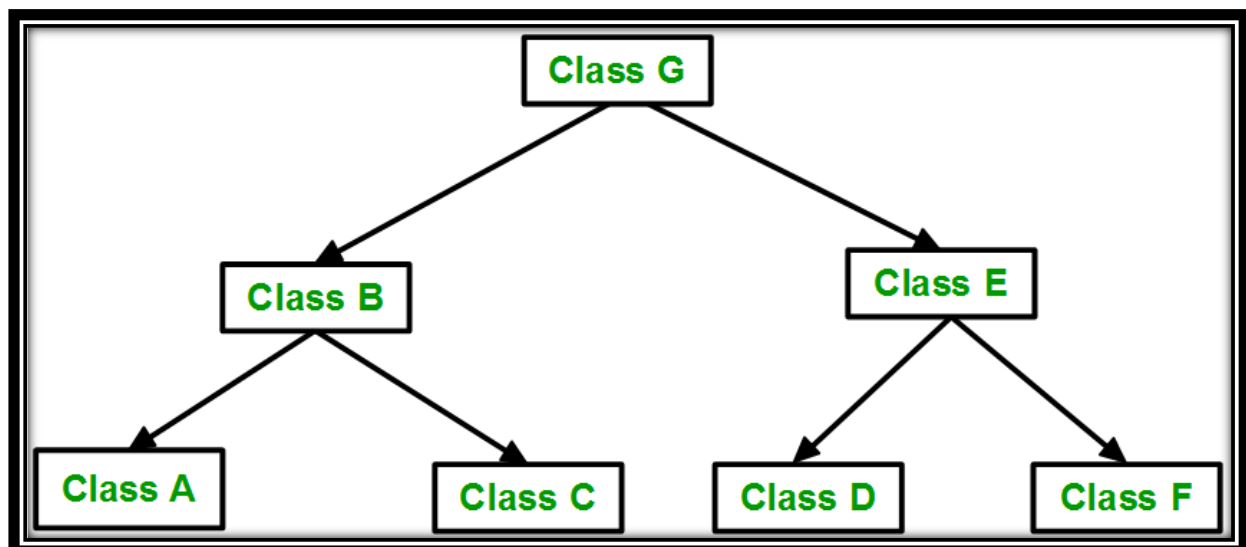
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

4) Hierarchical Inheritance:

- ✚ In this type of inheritance, more than one sub class is inherited from a single base class. i.e. more than one derived class is created from a single base class.



Example:

```
class vehicle
{
    public:
    vehicle()
    {
        cout<<"This is Vehicle";
    }
};
class car:public vehicle
{
    public:
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
        car()
        {
            cout<<"This is car";
        }
};
class bus : public vehicle
{
    public:
    bus()
    {
        cout<<"This is bus";
    }
};
void main()
{
    car c1;
    bus b1;
    getch();
}
```

5) Hybrid Inheritance:

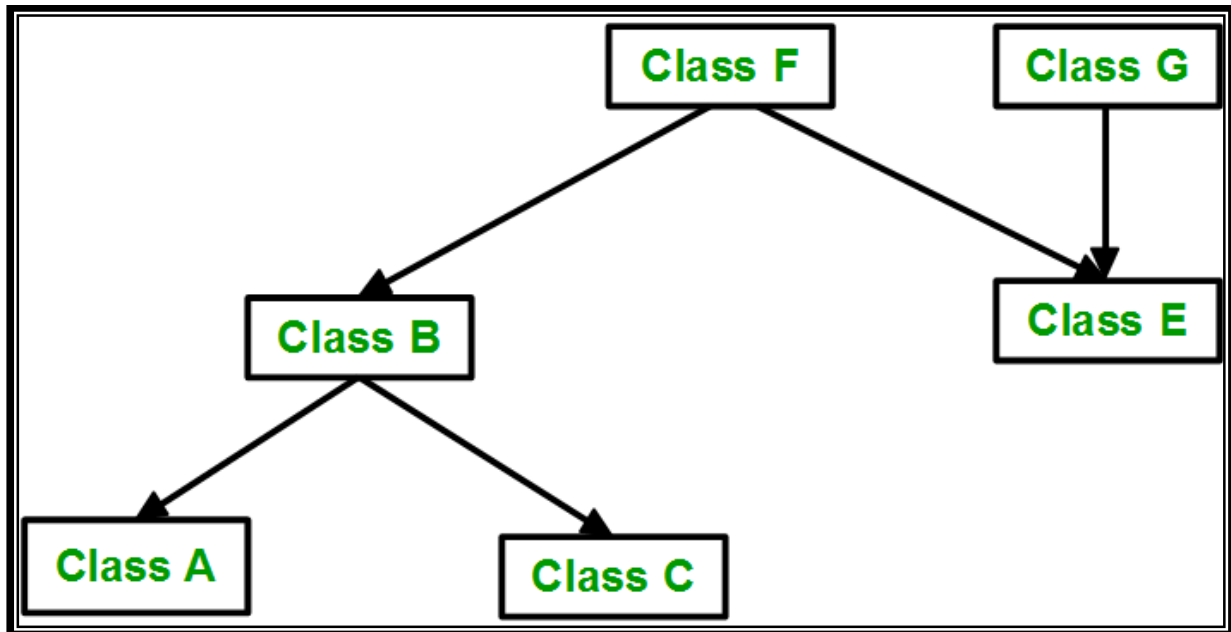
- Hybrid Inheritance is implemented by combining more than one type of inheritance. For example: Combining Hierarchical inheritance and Multiple Inheritance.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645



Example:

```
class vehicle
{
    public:
    vehicle()
    {
        cout<<"This is parent class";
    }
};
class fourwheeler:public vehicle
{
    public:
    fourwheeler()
    {
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
        cout<<"this is fourwheeler";
    }
};

class car:public fourwheeler, public vehicle
{

};

void main()
{
    car c1;
    getch();
}
```

M.C.Q		
Sr No.	Question	Answer
1.	In which type of inheritance there are multiple base class from which derived class can inherit?	Multiple
2.	In which type of inheritance, multiple child class is inherited from single base class?	Hierarchical
3. inheritance is the combination of any two types of inheritance	Hybrid
4.	In which type of inheritance, one class act as base class for other and other class act as base class for third class and so on?	Multilevel



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain Virtual Base Class in c++

Ans:

- Virtual base classes are used in virtual inheritance in a way of preventing multiple “instances” of a given class

Need for Virtual Base Classes:

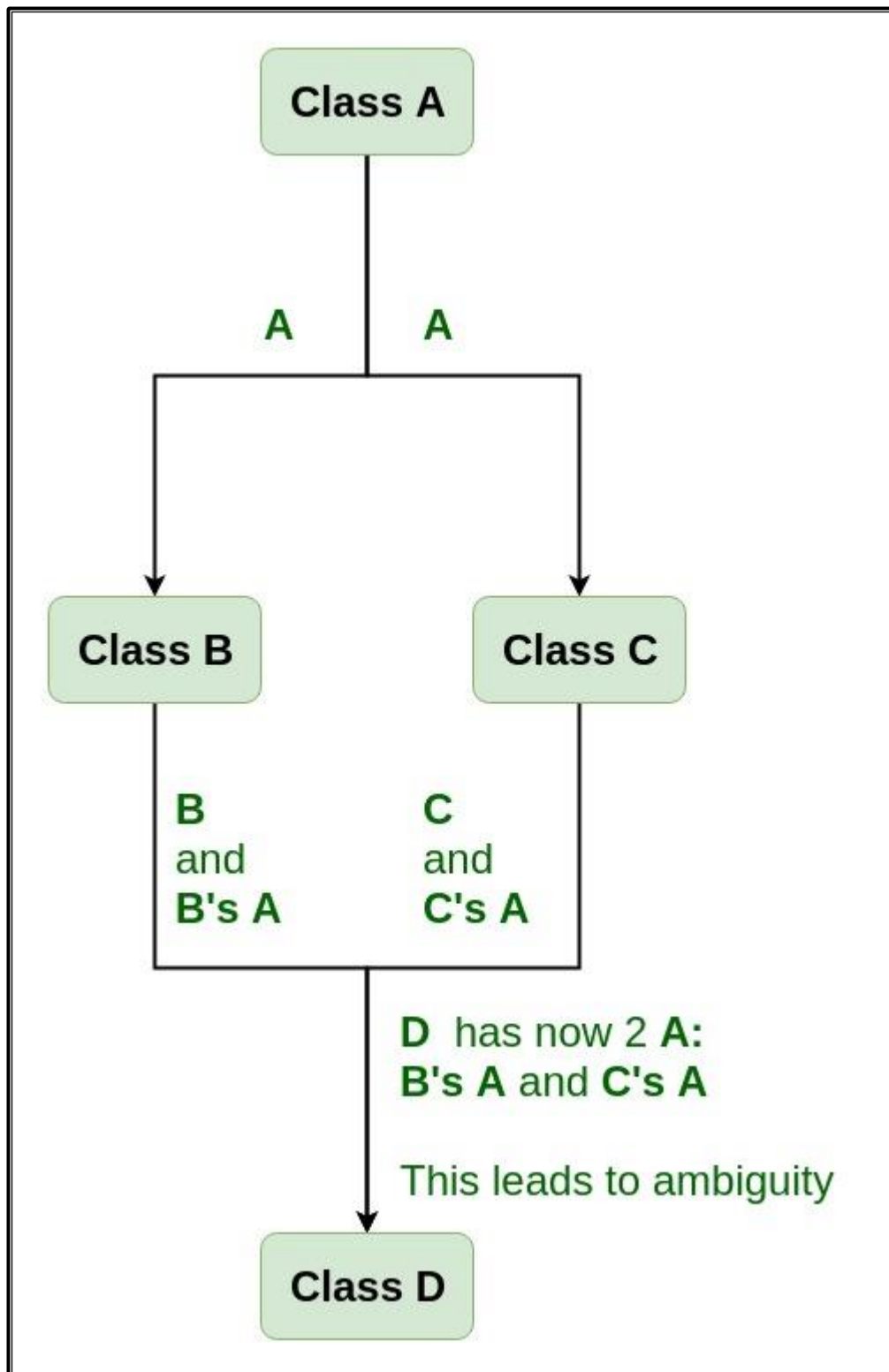
- Consider the situation where we have one class **A** .This class is **A** is inherited by two other classes **B** and **C**. Both these class are inherited into another in a new class **D** as shown in figure below.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645





SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- As we can see from the figure that data members/function of class **A** are inherited twice to class **D**. One through class **B** and second through class **C**. When any data / function member of class **A** is accessed by an object of class **D**, ambiguity arises as to which data/function member would be called? One inherited through **B** or the other inherited through **C**. This confuses compiler and it displays error.
- To resolve this ambiguity when class **A** is inherited in both class **B** and class **C**, it is declared as **virtual base class** by placing a keyword **virtual**

Syntax for Virtual Base Classes:

Syntax 1:

```
class B : virtual public A  
{  
};
```

Syntax 2:

```
class C : public virtual A  
{  
};
```

Note:

- **Virtual** can be written before or after the **public**. Now only one copy of data/function member will be copied to class **C** and class **B** and class **A** becomes the virtual base class.
- Virtual base classes offer a way to save space and avoid ambiguities in class hierarchies that use multiple inheritances. When a base class is specified as a virtual base, it can act as an indirect base more than once without duplication of its data members. A single copy of its data members is shared by all the base classes that use virtual base.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
class A
{
    public:
    void show()
    {
        cout<<"Hello";
    }
};

class B:public virtual A
{
};

class c:public virtual A
{
};

class D: public c, public B
{
};

void main()
{
    D d1;
    d1.show();
    getch();
}
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	To resolve ambiguity error the concept of virtual base class is used (T/F)?	True
2.	To declare class as virtual which keyword is used?	virtual
3.	virtual keyword can be written before or after the public (T/F)	True

Topic: Explain Abstract class

Ans:

- An abstract class is a class that cannot be instantiated and is usually implemented as class that has one or more pure virtual functions.
- A pure virtual function is the one which don't have any implementation, we can only declare it. A pure virtual function is declared by assigning 0 in declaration.
- The classes inheriting the abstract class must provide a definition for the pure virtual function; otherwise, the subclass would become an abstract class itself.

Example:

```
class vehicle
{
    public:
    virtual void wheel()=0;
};
class Bus:public vehicle
{
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
public:
void wheel()
{
    cout<<"This is child class";
}
};
void main()
{
    Bus b1;
    b1.wheel();
    getch();
}
```

M.C.Q		
Sr No.	Question	Answer
1.	We cannot create objects of abstract class or abstract class cannot be instantiated (T/F)?	True
2.	The functions which don't have any implementation, but only declaration are known as.....	Pure virtual function
3.	The pure virtual function is declared by	0
4.	The class containing pure virtual function is known as	abstract class
5.	The class inheriting abstract class must provide definition for	pure virtual function



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain Constructor in derived class.

Ans:

- A constructor is the special member function which initializes an object's data members and allocates memory at the time of creation of object.
- In inheritance, base class and derived class may have constructor. If base class has no argument constructor, then there is no need of constructor in derived class. However if the base class has constructor with parameter then it is mandatory for derived class to have a constructor and pass the arguments to the base class constructor.
- In inheritance, generally the object of derived class is created instead of base class. Hence the derived class has a responsibility to create a constructor and pass the arguments to the constructor of base class.
- When the object of derived class is created, the constructor of base class is executed first and later the constructor of derived class.

Example:

```
class base
{
    int i;
    public:
    base(int x)
    {
        i=x;
    }
    void show()
    {
        cout<<l;
    }
};
class derived:public base
{
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
int j;  
public:  
derived(int x,int y):base(x)  
{  
    j=y;  
}  
void show()  
{  
    base::show ();  
    cout<<j;  
}  
};  
  
void main()  
{  
    derived d1(5,6);  
    d1.show();  
    getch();  
}
```

M.C.Q		
Sr No.	Question	Answer
1.	If base class have default constructor then there is no need of constructor in child class (T/F)?	True
2.	If base class have parameterized constructor then it is compulsory for the child class to create a constructor (T/F)?	True



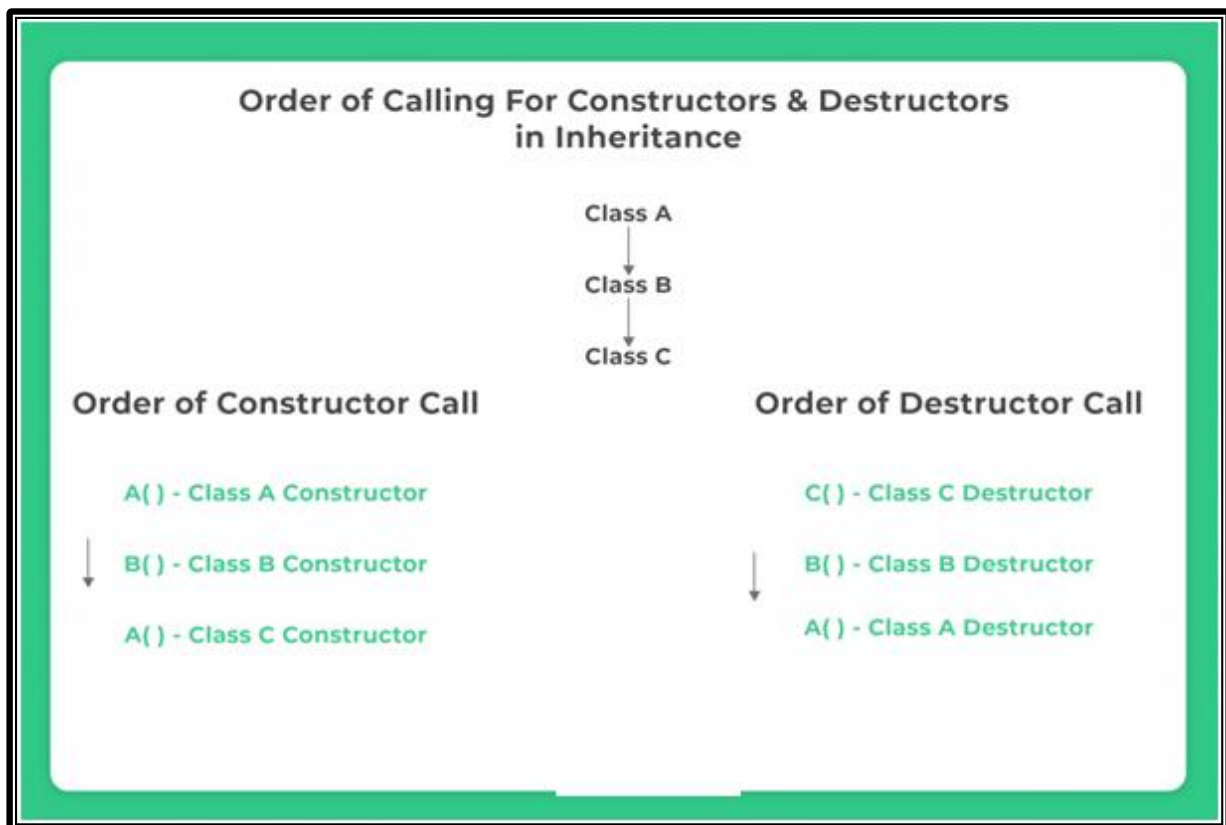
SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain Constructor and destructor in inheritance

Ans:



Example:

```
class parent
{
    public:
    parent()
    {
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
        cout<<"Parent class constructor called";
    }
    ~parent()
    {
        cout<<"Parent class destructor";
    }
};

class child: public parent
{
    public:
    child()
    {
        cout<<"Child class constructor";
    }
    ~child()
    {
        cout<<"Child Class destructor";
    }
};

void main()
{
    child c;
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	If we have base class constructor and child class constructor then which constructor is called first?	Base Class Constructor
2.	If we have base class constructor and child class destructor then which destructor is called first?	Child class

Topic: Explain Containership.

Ans:

- We can create an object of one class into another and that object will be a member of the class. This type of relationship between classes is known as containership or has_a relationship as one class contain object of another class.
- Containership is also known as composition.

Container Class:

The class which contains the object and members of another class is known as container class.

Contained Object:

The object that is part of another object is known as contained object.

Container Object:

The object that contains another object as its part is known as container object.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Inheritance	Containership
1) It enables a class to inherit the data and functions from base class	1) It enables a class to contain objects of different classes as its data members
2) The derived class may add data or functions to the base class	2) The container class cannot add anything to the contained class
3) Inheritance represents “is-a” relationship	3) Containership represents “has-a” relationship.

Note:

- 1) is-a relationship means one class is inheriting from another class
- 2) has-a relationship means one class has object of another class.

M.C.Q		
Sr No.	Question	Answer
1.	The class which contains object of another class is known as	Containership
2.	Containership is also known as	Composition
3.	Containership representsrelationship while inheritance representsrelationship	has-a is-a



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain Pointer to objects

Ans:

- Similar to the variables, objects can also have address and the pointer that can point to the address of the object is known as this pointer
- Every object in c++ has access to its own address through important pointer called this pointer
- This pointer is implicit parameter to all member functions
- Therefore, inside a member function, this may be used to refer to the invoking object.
- This pointer is not available in static member functions as static member functions can be called without any object.

Example:

```
class test
{
    int x;
    public:
    void setx(int a)
    {
        this->x=a;
    }
};
void main()
{
    test t1;
    test t2;
    t1.setx(10);
    t2.setx(20);
    getch();
}
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	Address of object is known aspointer	this
2.	this pointer refers to theobject	current
3.	this pointer is not available inmember functions	static

Topic: Write a short note on virtual and pure virtual functions

Ans:

Virtual Function:

- A virtual function is a member function which is declared within a base class and is re-defined (overridden) by a derived class.
- When you refer to derived class object using a pointer or a reference to the base class, you can call a virtual function for that object
- Virtual Functions are mainly used to achieve runtime polymorphism

Rules for Virtual Function

- 1) Virtual function cannot be static and also can not be a friend function of another class
- 2) Virtual functions should be accessed using pointer or reference of base class type to achieve runtime polymorphism
- 3) The prototype of virtual functions should be same in base as well as derived class.
- 4) Virtual Functions are always defined in base class and overridden in derived class.
- 5) Virtual class can have virtual destructor but it cannot have virtual



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

constructor

Note:

- If we have created virtual function in base class which is being overridden in child class then we don't have virtual keyword in the derived class.

Example:

class base

```
{  
    public:  
    virtual void print()  
    {  
        cout<<"Hello";  
    }  
};
```

class derived:public base

```
{  
    public:  
    void print()  
    {  
        cout<<"Hello";  
    }  
};
```

void main()

```
{  
    base *p1;
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
derived d;  
p1=&d;  
p1->print();  
getch();  
}
```

Pure Virtual Function:

- A pure virtual function is a virtual function for which we don't have an implementation , we only declare it
- A pure virtual function is declared by assigning 0 in the declaration.

Virtual Function	Pure Virtual Function
1) A virtual function is a member function of base class which can be redefined by derived class	1) A pure virtual function is a member function of base class whose only declaration is provided in base class, its implementation is done in derived class.
2) Classes having virtual function are not abstract class	2) Class having pure virtual function is known as abstract class.
3) The class having virtual function can be instantiated (that is you can create the object of the class containing virtual function)	3) The class having pure virtual function cannot be instantiated (that is you cannot create the object of the class containing pure virtual function)



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

M.C.Q		
Sr No.	Question	Answer
1.	The member function written in base class and overridden in child class is known as	Virtual Function
2.keyword is used to define virtual function	virtual
3.	The member function whose only declaration is done in base class and implementation is done in child class is known as	Pure Virtual Function
4.	The pure virtual functions are declared by assigning	0
5.	The class containing pure virtual function is known as	Abstract class.

Topic: Write a short note on default arguments to virtual functions

Ans:

```
class Base
{
    public:
        virtual void show(int x=0)
        {
            cout<<x;
        }
};
```

“C++”



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
class derived:public Base
{
    public:
    virtual void show(int x)
    {
        cout<<x;
    }
};

void main()
{
    derived d1;
    Base *b1;
    b1=&d1;
    b1->show();
    getch();
}
```

Note:

- 1) In the above example, show() is overridden in derived class. Default value is used at compile time. When compiler sees that an argument is missing in function call, it substitutes the default value given.
- 2) show() is called on base pointer. So the default value of base class will be substituted.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

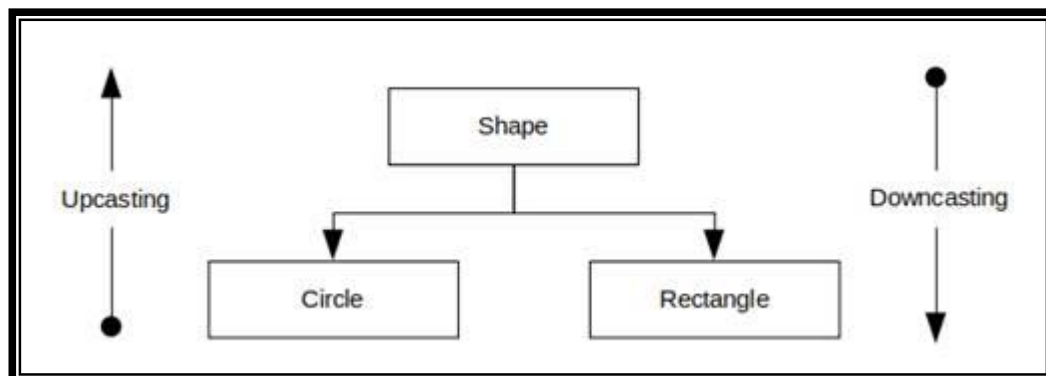
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Write a short note on RTTI

Ans:

- RTTI Stands for Runtime type Identification/Information.
- It is a mechanism that exposes information about an object's data type at runtime and is available only for the classes which have atleast one virtual function.
- It allows the type of an object to be determined during program execution.
- The simplest way to determine the runtime type of an object using a pointer or reference is to use the runtime cast that verifies that cast is valid.



Upcasting:

It is the process where derived class pointer is converted to base class pointer.

Downcasting:

It is the process where base class pointer is converted to derived class pointer.



SHREE H.N.SHUKLA GROUP OF COLLEGES
(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

SUBJECT: C++

CHAPTER-5:

File Handling

Sr No.	Topic
1.	C++ Streams
2.	C++ Stream Class
3.	Unformatted and Formatted I/O operations
4.	Opening and Closing a file
5.	Random Access File
6.	Command Line Arguments
7.	Error Handling Functions
8.	Templates
9.	Function Templates with multiple parameters
10.	Overloading of function template
11.	Non-type template arguments
12.	STL
13.	Differentiate bad() and fail()



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

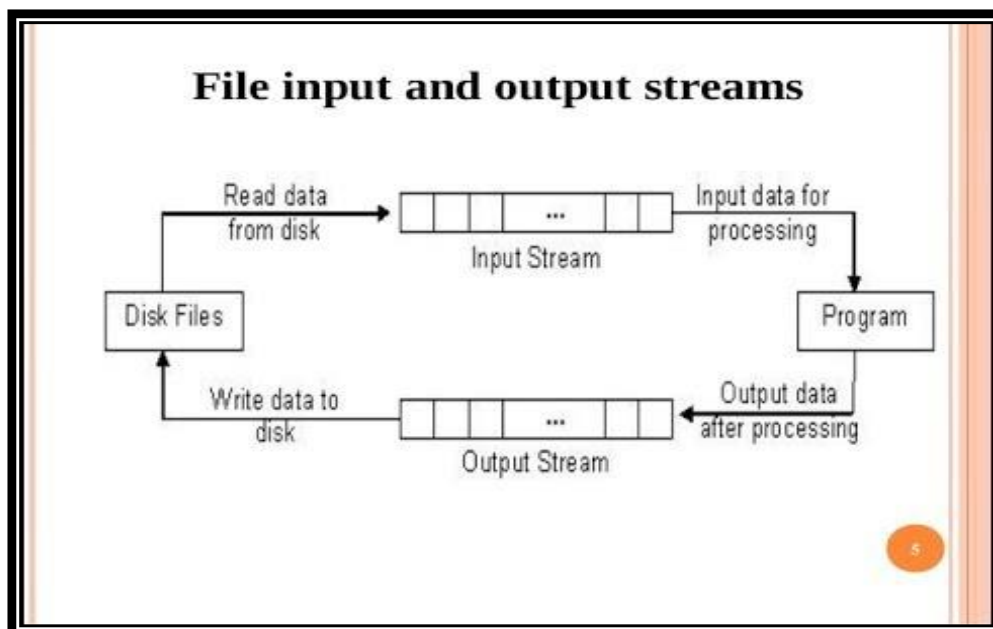
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: What is stream in c++?

Ans:

- ✿ In C++, stream refers to objects which are used to manipulate and interact with files. A stream is a series of bytes.
- ✿ It acts either as source from which input data can be extracted or as destination to which output data can be sent.
- ✿ The source stream provides data to the program called input stream and the destination stream that receive the data from the program is called output stream.





SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

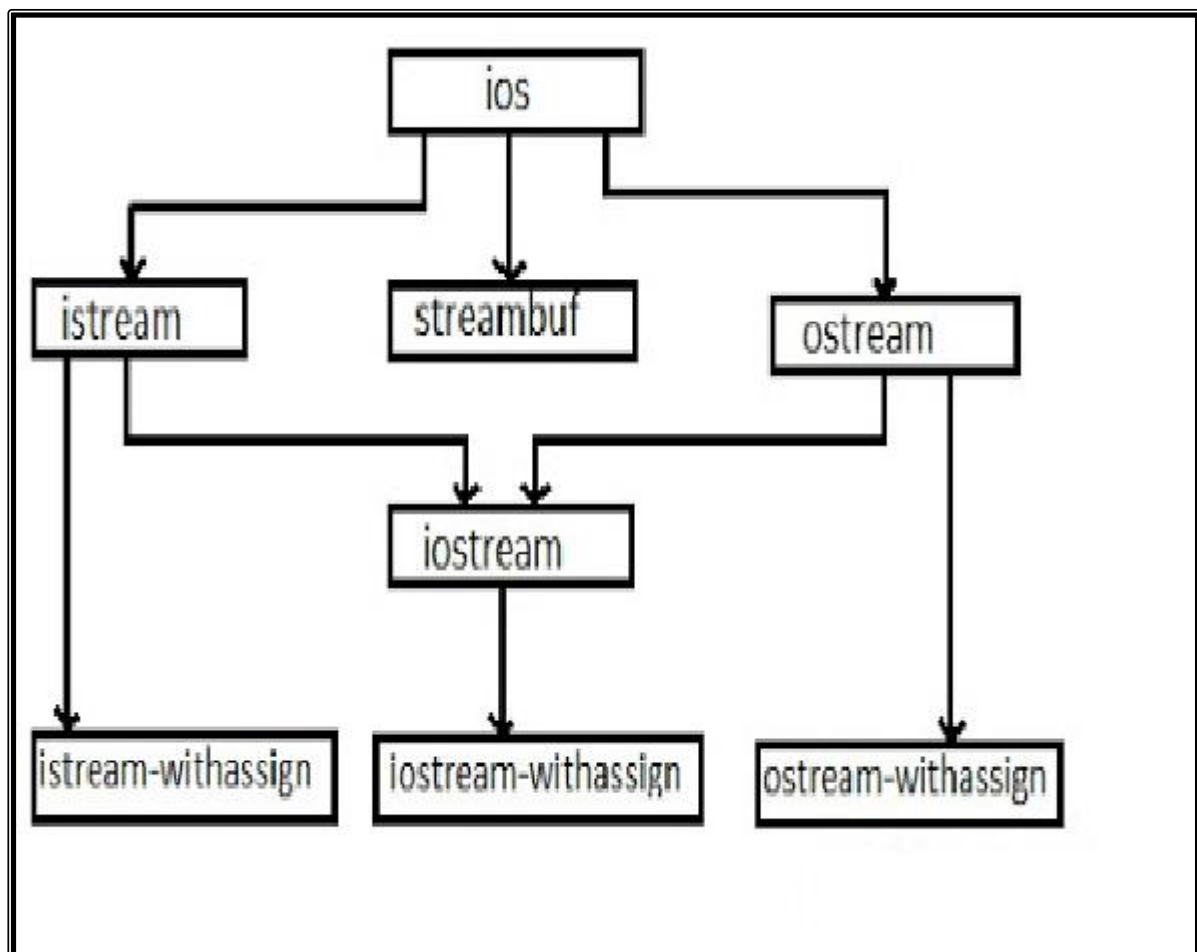
2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain C++ Stream Classes

Ans:

- 📌 C++ input-output system supports a hierarchy of classes that are used to manipulate both the input and output files. These classes are known as stream classes.





SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

1. **The ios class:** The ios class is responsible for providing all input and output facilities to all other stream classes.
2. **The istream class:** This class is responsible for handling input stream. It provides number of function for handling chars, strings and objects such as get, getline, read, ignore, putback etc..

Example:

```
void main()
{
    char x;
    cin.get(x);
    cout<<x;
    getch();
}
```

3. **The ostream class:** This class is responsible for handling output stream. It provides number of function for handling chars, strings and objects such as **write, put** etc..

Example:

```
void main()
{
    char x;
    cin.get(x);
    cout.put(x);
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

4. **The iostream class:** This class is responsible for handling both input and output stream as both istream class and ostream is inherited into it. It provides functions of both istream class and ostream class for handling chars, strings and objects such as get, getline, read, write etc.

Example:

```
void main()
{
    char x;
    cin.get(x);
    cout.write("hello"5);
    getch();
}
```

5. **istream_withassign class:** This class is variant of istream that allows object assignment. that is cin is the object of this class and may be reassigned at runtime to different istream object.
6. **ostream_withassign class:** This class is variant of ostream that allows object assignment that is cout is the object of this class and may be reassigned at runtime to different ostream object.

Topic: Explain unformatted and formatted I/O Operations

Ans:

Unformatted I/O Operations:

- 1) **get():** It is the member function of input stream class istream. It is used to read single character from the input device.

Example:

```
void main()
{
    char ch;
    while(ch!=\n')
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
    {  
        cin.get(ch);  
        cout<<ch;  
    }  
}
```

- 2) put(): This function is the member of ostream and it is used to output the line of text, character by character

Example:

```
void main()  
{  
    char x;  
    cin.get(x);  
    cout.put(x);  
    getch();  
}
```

- 3) getline(): This function is used to read the whole line of text or until the maximum is reached.

Example:

```
void main()  
{  
    char s[20];  
    cout<<"Enter the string";  
    cin.getline(s,20);  
    cout<<"string is"<<s;  
    getch();  
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Note: cin function is also used to read the string but it can not read the string after white space character. It will truncate the part after white space character

4) write (): This function displays the entire line on the screen.

Example:


```
void main()
{
    char s[20];
    cout<<"Enter the string";
    cin.getline(s,20);
    cout.write(s,20);
    getch();
}
```

Formatted I/O Operations:

- C++ I/O system allows formatted I/O operations. For example, how many digits after decimal point will be displayed
- C++ supports a wide variety of features to perform input or output in different formats.
- These features include:
 - 1) ios stream class member functions
 - 2) standard manipulators
 - 3) user-defined manipulator

1) ios stream class member function:

 Following are the functions defined in ios class.

 **width():** This method is used to set the width of required field.

Example:



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
void main()
{
    char ch='A';
    int temp=10;
    cout<<ch;
    cout.width(10);
    cout<<temp;
    getch();
}
```

➤ **precision():** This method is used to set the number of decimal point to float value.

Example:

```
void main()
{
    cout.precision(2);
    cout<<3.1425;
    getch();
}
```

➤ **fill():** This method is used to set a character to fill in the blank space of a field.

Example:

```
void main()
{
    char ch='A';
    clrscr();
    cout.fill('*');
    cout.width(10);
    cout<<ch<<"\n";
}
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
getch();
```

```
}
```

2) Use of Manipulators:

- ✿ The manipulators are the functions that can be used with << or >> operator to alter the behaviour of any stream class instances including cout and cin.
- ✿ All the predefined manipulators are defined in iomanip.h
- ✿ There can be more than 1 manipulators in same statement.
- ✿ The manipulators are categorized into two types:
 - 1) Non-parameterised manipulators
 - 2) Parameterized manipulators

1) Non-parameterised manipulators:

Following are the types of non-parameterised manipulators:

Manipulator	Action Performed
ws	Extracts white space characters from input stream characters
endl	Outputs a newline (“\n”)
ends	Outputs a null character (“\0”)
flush	Flushes the stream

2) Parameterised manipulators:

Following are the types of non-parameterised manipulators:

Manipulator	Action Performed
setw(int width)	Sets the field width
setprecision(int prec)	Sets the floating point precision
setfill (int fchar)	Sets the fill character



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Example:

```
#include<iomanip.h>
void main()
{
    float f1=124.46;
    clrscr();
    cout<<setfill('#')<<setprecision(2);
    cout<<endl<<setw(10)<<f1;
    getch();
}
```

3) User-defined Manipulators:

- C++ I/O stream is customized by creating user defined manipulator.
- The user can also design their customized manipulators to control the appearance of the output depending upon their requirements

Topic: Opening and closing of a file

Ans:

• Some of the standard file handling classes are:

- 1) **Ofstream:** This file handling class in C++ signifies the output file stream and is applied to create files for writing information to files.
- 2) **ifstream:** This file handling class in C++ signifies the input file stream and is applied for reading information from files.
- 3) **fstream:** This file handling class in C++ signifies the file stream generally, and has the capabilities for representing both ofstream and ifstream

Opening a File:

- Opening a file is the first operation performed on an object.
- File can be opened in 2 ways:
 - 1) Using constructor function of the class



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

2) Using member function open of the class

Syntax:

open (filename, mode);

✚ Following are the different modes of file opening:

- 1) ios::app : Append mode
- 2) ios::ate : sets the stream position indicator to the end of the stream before each output operation
- 3) ios::in : Read mode
- 4) ios:: out: Write mode
- 5) ios::trunc: Truncated mode

✚ Functions used for file handling are:

- 1) open(): To create a file
- 2) close(): To close an existing file
- 3) get(): to read a single character from the file
- 4) put(): to write a single character in the file
- 5) read(): to read data from a file
- 6) write(): to write data into a file

Closing a File:

✚ When any c++ program terminates, it automatically flushes out all the streams releases the allocated memory and closes all the opened files.

✚ When the file is opened, it should be closed

Syntax:

close (filename);

Example 1:

```
#include<fstream.h>
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
void main()
{
    ofstream f;
    f.open("1.txt",ios::out);
    f<<"Writing";
    f.close();
    getch();
}
```

Example 2: Write a program to write into file and read that data and display on the screen

```
#include<fstream.h>

void main()
{
    ifstream f1;
    f1.open("bscit.txt");
    if(f1==NULL)
    {
        cout<<"Can not open File";
    }
    else
    {
        while(f1.eof()==0)
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
{  
    f.get( c);  
    cout<<c;  
}  
  
f1.close();  
getch();  
}
```

Example 3: Write a program to copy data from one file to another file.

```
#include<fstream.h>  
  
void main()  
{  
  
    ifstream f1;  
    ofstream f2;  
    f1.open("bscit.txt");  
    if(f1==NULL)  
    {  
        cout<<"Can not open File";  
    }  
    else  
    {  
        while(f1.eof()==0)  
        {  
            f.get( c);  
            f2<<c;  
        }  
  
        f1.close();  
        f2.close();  
        getch();  
    }  
}
```

Example 4: Write a program to write odd numbers in "odd.txt" and even numbers in "even.txt"



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
#include<fstream.h>

void main()
{
    ifstream f1;
    ofstream f2,f3;
    f1.open("data.txt");
    if(f1==NULL)
    cout<<"Can't open the file;
    f2.open("odd.txt");
    f3.open("Even.txt");
    int n;
    f1>>n;
    while(f1)
    {
        if(n%2==0)
        {
            cout<<"Even";
            f2<<n;
        }
        else if(n%2!=0)
        {
            cout<<"Odd";
            f3<<n;
        }
        f1>>n;
    }
    f1.close();
    f2.close();
    f3.close();
    getch();
}
```

Topic: Write a short note on random access file:

Ans:

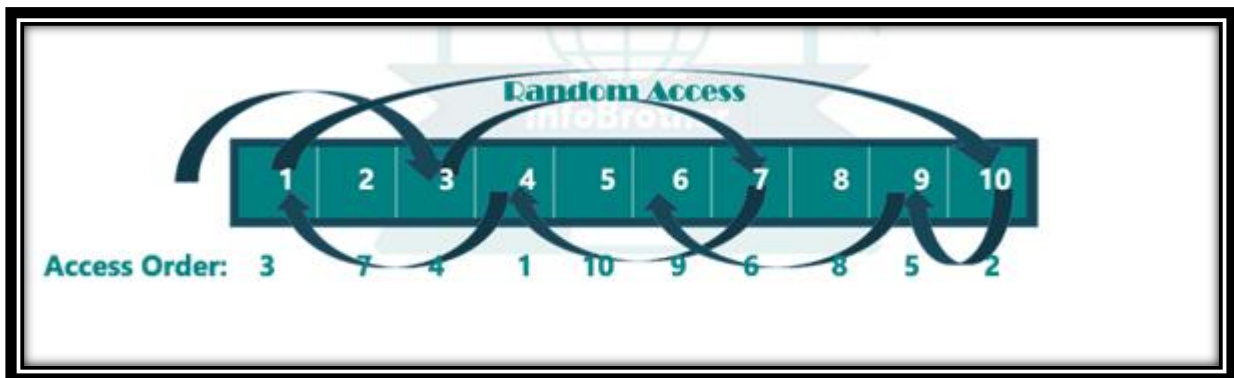


SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- ✿ Unlike Sequential files, we can access Random Access Files in any order we want.
- ✿ Example: Songs on record: We can go directly to any song we want without having to play or fast-forward over the songs. That is the order of play has nothing to do with the order of in which songs were originally recorded.



- ✿ C++ I/O system supports several functions for setting a file pointer to any desired position inside the file or gets the current file pointer.

Following are the functions:

- 1) seekg() : Moves get file pointer to a specific location. It is member of ifstream
- 2) seekp(): Moves put file pointer to a specific location. It is the member of ofstream
- 3) tellg(): Returns the current position of get pointer. It is the member of ifstream.
- 4) tellp(): Returns the current position of put pointer. It is the member of ofstream.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

seek() takes two arguments:
seekg(offset,refposition);
seekp(offset, refposition);

■ reposition means:

- 1) ios::beg: seek from beginning of file
- 2) ios::cur: seek from the current location
- 3) ios::end: seek from the end of file

Seek Call	Action Performed
out.seekg(0, ios::beg)	Go to the beginning of the file
out.seekg(0,ios::cur)	Stay at the current location
out.seekg(0, ios::end)	Go to the end of file
out.seekg(m,ios::beg)	Move to (m+1) byte location in the file
out.seekg(m,ios::cur)	Move forward by m bytes from current position
out.seekg(-m,ios::cur)	Move backward by m bytes from current position
outseekg(-m,ios::end)	Move backward by m bytes from the end
in.seekp(-m,ios::cur)	Move write pointer backward by m byte

Example:

```
#include<fstream.h>
void main()
{
    ifstream f1;
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
clrscr();
f1.open("HT.txt");
if(f1==NULL)
cout<<"Can not open the file";
else
{
    cout<<f1.tellg();
    f1.seekg(5,ios::beg);
    cout<<f1.tellg();
    char c=f1.get();
    cout<<c;
    cout<<f1.tellg();
    f1.seekg(-5,ios::end);
    cout<<f1.tellg();
    c=f1.get();
    cout<<c;
}
f1.close();
getch();
}
```

Topic: Write a short note on command line arguments

Ans:

- ✚ It is possible to pass some values from the command line when they are executed. These values are known as command line arguments.
- ✚ The command line arguments are handled using main() arguments where argc refers to the number of arguments passed, and argv[] is a pointer which points to each argument passed to the program.
- ✚ In command line arguments, argv[0] means the first argument which



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

will denote name of the program...argv[1] means the second argument, argv[2] means third argument.

Example:

```
#include<iostream.h>  
#include<conio.h>
```

```
void main(int argc, char *argv[])  
{  
    if(argc!=2)  
        cout<<"ArgumentNotGiven"  
    else  
        cout<<argv[1];  
        getch();  
}
```

Topic: Write a short note on Error Handling in c++

Ans:

- ✿ An exception is a problem that arises during the execution of a program.
- ✿ A c++ exception is a response to an exceptional circumstance that arises while a program is running such as an attempt to divide by zero.
- ✿ Exceptions provides a way to transfer the control from one part of program to another
- ✿ C++ exception is built on 3 keywords:
 - 1) try
 - 2) catch
 - 3) throw



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- 1) try: A try block identifies a block of code for which particular exceptions will be activated. It is followed by one or more catch blocks.
- 2) catch: A program catches an exception with an exception handler at the place in program where you want to handle the problem.
- 3) throw: A program throws an exception when problem shows up.

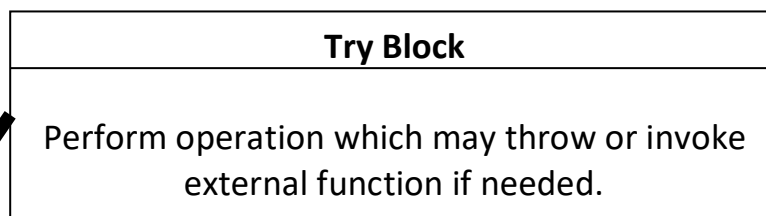
Advantages:

1) **Separation of Error handling code from Normal Code:**

In traditional error handling codes, there are always if else conditions to handle errors. These conditions and the code to handle the error get mixed up with normal flow. This makes code readable and maintainable. While try catch blocks, the code for error handling becomes separate from normal flow.

2) **Functions can handle any exceptions:** A function can throw many exceptions, but may choose to handle some of them.

3) **Grouping of error types:** In C++, both basic types and objects can be thrown as exception.



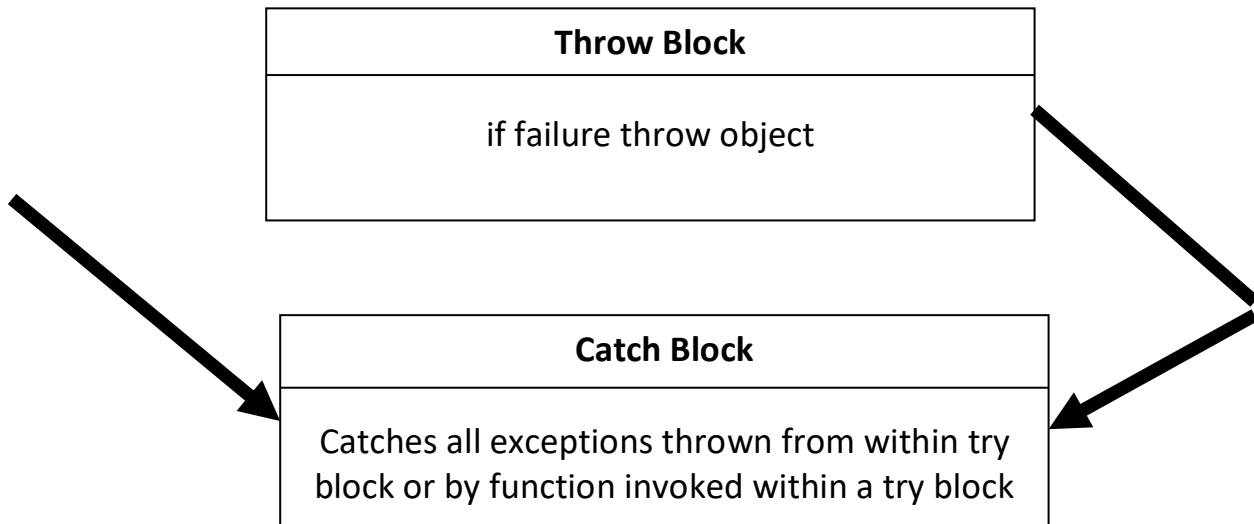
"C++"



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645



Example:

```
int main()
{
    int a=5,x;
    cout<<"Enter x";
    cin>>x;
    try
    {
        if(x==0)
            throw x;
        else
            cout<<(a/x);
    }
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
        catch(int i)
        {
            cout<<"Exception caught";
        }

        return 0;
    }
```

Topic: Explain templates

Ans:

- ✚ Classes can also be declared to operate on different data types, such classes are known as class templates.
- ✚ It specifies a generic class which uses logic that can be generalized.
- ✚ Generic class is the class that allows built in and user defined data types to be a parameter to methods, classes
- ✚ For example, classes like an array, map etc. which can be used using generics very efficiently.

- ✚ Templates can be represented in 2 ways:

- 1) Function templates
- 2) class templates

1) Function Templates:

- ✚ We can define a template for the function. For example, if we have add(), we can create versions of add function for adding, int, float or double type values.
- ✚ A function template is created by using the keyword template.
- ✚ The template defines what function will do.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Syntax:

```
template <class Ttype> returntype functionname(parameter list)
{
}
}
```

- 🌀 In the above syntax, Ttype is the placeholdername for a data type used by the function. A class keyword is used to specify a generic type in template declaration.

Example:

```
template <class T>T add(T x, T y)
{
    int result;
    result=x+y;
    cout<< result;
}

void main()
{
    int i=10;
    int j=20;
    float a=1.2;
    float b=2.2;
    add(i,j);
    add(a,b);
    getch();
}
```



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Function template with multiple parameters:

Ans:

- Function template can also contain multiple parameters

Syntax:

```
template <class T1, class T2, .....>
{
}
}
```

Example:

```
template <class X, class Y>void fun(X a, Y b)
{
    cout<<"Value of a is"<<a;
    cout<<"Value of b is"<<b;
}

void main()
{
    fun(15,2.5);
    return 0;
}
```

2) Class Template:

- We can define a template for the class.
- When the class uses the concept of template, then class is known as generic class.



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

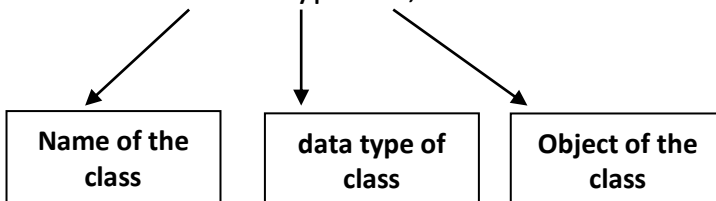
Syntax:

```
template <class Ttype>  
class classname  
{  
    ....  
    ....  
}
```

In the above syntax, Ttype is the placeholder name which will be determined when the class is instantiated.

Creating object of the class:

```
classname <type>ob;
```



Example:

```
template <class T>  
class A  
{  
    public:  
    T num1=5;  
    T num2=6;  
    void add()  
    {  
        cout<<"Addition is"<<num1+num2  
    }  
};
```




SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

```
int main()
{
    A<int>d;
    d.add();
    return 0;
}
```

Topic: Explain overloading of function templates

Ans:

- Function template overloading means function template with same name but different parameters.

Example:

```
template <class X>void fun(X a)
{
    cout<<"Value of a is"<<a;
}
template <class X, class Y>void fun(X b, Y c)
{
    cout<<"Value of b is"<<b;
    cout<<"Value if c is"<<c;
}
int main()
{
    fun(10);
    fun(10,11);
    return 0;
}
```

"C++"



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

Topic: Explain Non-type template argument

Ans:

- ✚ A template non-type parameter is a special type of parameter that does not substitute for a type, but is instead replaced by a value.
- ✚ A template can have multiple arguments.
- ✚ A non-type parameter may be either of following: integers, strings, function names, constant expression and built-in types.

Example:

```
template <class T,int size>  
class array  
{  
    T a[size];  
  
};
```

```
array<int,10>; //Array of 10 integers  
array<float,5> // Array of 5 floating point numbers
```

Topic: Write a short note on STL

Ans:

- ✚ The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc.
- ✚ STL has four components:
 - 1) Algorithms
 - 2) Containers
 - 3) Functions



SHREE H.N.SHUKLA GROUP OF COLLEGES (Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

4) Iterator

- 1) **Algorithms** : They provide means by which you can perform initialization, sorting, searching and transforming the contents of container,
- 2) **Container**: It stores object and data.
 - 🌿 sequence Containers: Implement the data structure which can be accessed in a sequential manner
 - 1) Vector
 - 2) array
 - 3) List
 - 🌿 Container Adapters: Provide a different interface for sequential containers.
 - 1) Queue
 - 2) stack
 - 🌿 Associative Containers: Implement sorted data structure that can be quickly searched.
 - 1) map
 - 2) set
 - 🌿 Unordered Associative Containers: Implement unordered data structures that can be quickly searched.
- 3) **Functions**: STL includes classes that overload the function call operator. Instances of such classes are known as objects of **functors**. **Functors** allows the working of the associated functions to be customized with the help of parameters to be passed.
- 4) **Iterator**: Iterators are used for working upon a sequence of values. They are primarily used in sequence of numbers, characters etc. Following are the operations of iterators:
 - 1) begin



SHREE H.N.SHUKLA GROUP OF COLLEGES
(Affiliated to Saurashtra University and G.T.U)

2 – Vaishalinagar,
Nr.Amrapali Railway crossing
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2440478

3 – Vaishalinagar,
Nr. Amrapali Railway crossing,
Raiya Road,
Rajkot – 360001.
Ph No. 0281- 2471645

- 2) end
- 3) advance
- 4) next
- 5) prev

Topic: Differentiate fail() and bad()

Ans:

fail()	bad()
1) It returns TRUE if read or write operation has failed otherwise false	1) It returns true if an invalid operation is attempted or any unrecoverable error has occurred.