Important Short Questions

Q2. Give short answer of the following questions.

Q1. What is the main purpose of C++ programming?

Ans. the main purpose of C++ programming is to add object orientation to the C++ programming language classes are the central feature of C++ that supports object oriented programming and are often called us defined types.

Q2. What is the most important feature of C++?

Ans. The classes are the most important feature of C++ that leads to Object Oriented Programming.

Q3. Define class.

Ans. Class is a user defined data type which holds its own data members and members functions.

Q4. What is meant by object?

Ans. Classes can be accessed and used by creating instance of the class which is called object.

Q5. State the format of class.

Ans. The general format is

Class class_name

```
{
Access_specifier_1       // body of the class
Member 1:
Access_specifier_2:
Member 2:
};
```

Q6. The body of class consists of how many parts?

Ans. the body of the class consists of

1. Data members
2. Member functions

Q7. Define access specifier.

Ans. data members and member functions are used within the access specifiers

Q8. Name the types of access specifier.

Ans. an access specifier is one of the three types of

1. Private
2. Public
3. Protected

Q9. Define instance of a class.

Ans. In C++ when we declare a variable of type class we call it instantiating (from instance) the call and variable itself is called an instance or object of that class.

Q10. State the importance of Object in OOP.

Answer

Object is of great importance in OOP because a class cannot be used without creating its object

Q11. State the importance of Object in OOP.

Ans. the general syntax for creating an object of a class is given below

Class_name Object_name;

Q12. How many members of a class have?

Ans. class has two members

1. Data members
2. Member functions

Q13. What does data member mean?

Ans. the attributes of a class are called data members.

Q14. Where we can declare data members in program?

Ans. Mostly data members are declared under the private access specifier to make them private to the class which they are used. Data members can also be used under public access specifier

Q15. What is meant by member function?

Ans. the functions declared or defined inside the body of the class are called member functions

Q16. Where we can declare data members in program?

Ans. Member functions are usually written under the public access specifier to operate on the data members the class. Member functions can also be written in the private area of the class but the practice is to write them the public area.

Q17. What do access specifiers in C++ mean?

Ans: access specifiers in C++ define how the members of the class can be accessed. Access specifiers determine which member of a class is accessible in which part of the class/program

Q18. What are the most commonly used access specifiers in C++?

Ans. the most commonly used access specifiers in C++ are given below

1. **Private access specifiers**
2. **Public access specifiers**

Q19. Define private access specifiers

Ans. Private access specifiers tells the compiler that the members defined in a class by preceding this specifier are accessible only within the class and its friend function. Private members are not accessible outside the class.

Q20. Define public access specifier.

Ans. public members are accessible from anywhere where the object is visible i.e within the class in the derived classes and in the main function.

Q21. What is meant by data hiding?

Ans. data hiding is one of the C++ features in which the members of a class are protected against illegal access from outside the class.

Q22. Name the different access level in class.

Ans. in C++ we have the facility of hiding data using different access levels in class

1. Private
2. Protected
3. Public

Q23. Make a table to show the level of hiding members of a class

Ans. the following table shows the level of hiding members of a class

Access	Public	Protected	Private
Access of members (data functions) in the same class	Yes	Yes	Yes
Access of members (data functions) in the derived class	Yes	Yes	No
Access of members (data functions) from outside the class	Yes	No	No

Q24. Define constructor.

Ans. a constructor is a special type of member function that initializes an object automatically when it is created. Compiler identifies a given member function is a constructor by its name and return type.

Q25. Write down the rules for naming constructor.

Ans. unlike functions, constructor have specific rules/features for how they must be names:

1. Constructors have the same name as that of the class
2. Constructors have no return type
3. Constructor is always public

Q26. Name the types of constructors

Ans. following are the two types of constructors

1. Default constructor/ implicit constructor
2. User-defined constructor/explicit constructor

Q27. Define default constructor

Ans. a constructor without any arguments or with default values for every arguments is treated as default/implicit constructor. If we do not declare any constructors in a class definition the compiler assumes the class to have a default constructor with no arguments.

Q28. What does constructor-overloading mean?

Ans. Using more than one constructor in the same class is called constructor-overloading.

Q29. What are destructors?

Ans. Destructors are special member functions that are exerted when an object is destroyed. Destructor fulfils the opposite functionality of constructor and thus de-allocates the memory already allocated to an object during its creation ~ destructors denotes them.

Q30. State the features of destructor.

Ans. destructors have the following special features

1. Destructors has the same name as that of the class, preceded by a symbol ~
2. Destructor cannot take arguments
3. Destructor has no return type

Q31. What is the method to access data members?

Ans. to access members of an object, whether data members or member functions, dot operator (.) is used with the member name as shown here under

Object.member_name;

Q32. What is the method to access member function?

Ans. member functions of a class can be accessed by the same way as data members are accessed i.e by the use of dot operator (.) in concatenation with the member function name as given below

Object.member function_name;

Q33. What does parent class mean?

Ans. a parent class is the class from which other classes are derived.

Q34. What does child class mean?

Ans. a sub class is a class which inherits features from the base class. A sub-class is also called child class.

Q35. Write down the syntax of using inheritance in classes.

Ans. syntax of using inheritance in classes

```
Class A          //base class
{
};
Class B    public A          //class B is derived from class A
{
};
```

Q36. Define multiple polymorphism.

Ans. in inheritance, sometimes a class is derived from more than one parent class and the phenomenon is called multiple inheritance.

Q37. Define polymorphism.

Ans. polymorphism is the ability to use an operator or function in multiple ways. Polymorphism gives different meanings or functionality to the operators or functions.

Q38. What does poly refers.

Ans. Poly refers to many, signifies that there are many uses of these operators and functions. It is the use of a single function or operator in many ways.

Q39. How many methods are there to perform polymorphism?

Ans. following are the methods

1. Function name overloading
2. Operator overloading
3. Virtual functions

Q40. What does function-overloading mean?

Ans. function overloading is the concept of using name function name for related but slightly different purposes in a single program

Q41. What does operator-overloading mean?

Ans. a single operator + behaves differently in different contexts such as integer and float addition and strings concatenation. This concept is known as operator overloading.

Q42. What is meant by virtual function? Why it is important?

Ans. a virtual function is a function or method whose behavior can be overridden within an inheriting class by a function with the same signature. This concept is a very important part of the polymorphism portion of OOP

Q43. Give an example of inheritance from daily life.

Ans. a new model car having some inherited features from the old model like break system and navigation system etc.

Q44. Give an example of polymorphism from daily life.

Ans. a person can be a student as well as a friend to another person. Also a person can be an engineer as well as a teacher.

Q45. Write a program to illustrate the working of objects and class in C++ programming.

Ans. .....

Q46. By using constructor, write a program in C++ to calculate the area of a rectangle and display it.

Ans. program

```
#include <iostream>
Using namespace std;
Class Area
{
Private:
Int length;
Int breadth;
Public:
//constructor
Area ( ): length (5), breadth (2) { }
Void getlength ( )
{
Cout < < "Enter length and breadth respectively";
Cin > > length > > breadth;
```

```
}
Int AreaCalculation ( ) { return (length * breadth): }
Void DisplayArea (int temp)
{
Cout < < "Area : " << temp:
}
};
Int main ( )
Area A1, A2:
Int temp:
A1. Getlength () ;
Temp=A1. AreaCalculation ( );
A1. DisplayArea (temp):
Cout < < end1 < < "Default area when value is not taken from user " < < end1;
Temp=A2. AreaCaculation ( ):
A2.DisplayArea (temp):
Return 0:
}
```

Q47. Write a program in C++ to calculate the area of a rectangle and display it by using overloaded constructor.

Ans. Program

```
#include <iostream>
```

```
Using namespace std;
Class Area
{
        Private:
                Int length;
                Int breadth;
Public:
// Constructor with no arguments
Area () : length (5), breadth (2) { }
Void (int l, int b): length (1) , breadth (b) { }
Void Get Length ()
{
        Cout << "Enter length and breadth respectively: "
        Cin >> length >> breadth;
    }
    Int AreaCalculation () { retrn length * breadth; }
    Void Display Area (int temp)
    {
        Cout<< "Area: " << temp << end1;
    }
};
Int main ()
{
```

```
Area A1, A2, (2, 1)

Int temp:

Cout << "Default Area when no argument is passed" << end1:

Temp = A1. AreaCalculation ()':

A1. DisplayArea (temp):

    A2. DisplayArea (temp):

Return 0:

}
```

Q48. Write a C++ program to add two complex numbers by passing objects to a function.

Ans. Program

```
#include <iostream>

Using namespace std;

Class Complex

{

Privae:

        Int real;

        Int imag;

Public:

    Complex (): real (0), imag (0) { }

Void readData ()

{
```

```
Cout << "Ente real and imaginary number respectively:"<<end1;
Cin >> real >> imag;
}
Void addComplex Numbers (Complex comp1, Complex comp 2)
{
// real represents the real data of object c3 because this function is called using
        Code c3. Add (c1, c2)
Real = comp1, real+ comp2, real;
// imag represents the imag data of object c3 because this function is called using
Imag = comp1. Imag+comp2, imag;
}
Void display Sum ()
{
Cout << Sum = << real << "+" << imag << "I";
}
};
Int main ()
{
        Complex c1, c2, c3;
        C1. readData ();
        C2. readData ();
        C3. addComplex Numbers (c1, c2);
        C3. Display Sum ();
```

```
Return 0:
}
```

Q50. Write a program in C++ to explain the concept of operator overloading in C++ programming

Ans. Program

```
#include <iostream>
Using namespace std:
Class Test
{
        Private:
                Int count:
        Public:
        Test (): count (5){}
        Void operator ++ {}
        {
                Count = count + 1:
        }
        Void display () { cout <<"count: "<<count:}
};
Int main ()
{
        Test t:
```

```
        // this calls "function void operator ++ ()" function
        ++ t;
        t. Display ();
        return 0;
}
```

Q51. Write a C++ program to add two numbers using classes

Ans. Program

```
#include <iostream>
Using namespace std;
// class definition
Class Numbers
{
Private :
        Int a;
        Int b;
Public:
        // member function declaration
        Void readNumber (void);
        Void printNumbers (void);
        Int calAddition (void);
}
// member function definitions
```

```
Void numbers : read numbers (void)
{
        Cout <<"Enter first number: "
        Cin>>a;
        Cout <<"Enter second number":
        Cin >> b;
}
Void numbers  : : print Numbers (void)
{
        Cout <<"a= "<<a<<",b="<<b<<end1;
}
Int numbers : : calAddition (void)
{
        Return (a + b);
}
// main function
Int main ()
{
        //declaring object
        Number num;
        Int add; // Variables to store addition
        // take input
        Num. readNumbers ();
```

```
// +ind addition

Add=num. calAddition ():

// print addition

Cout << "Addition/sum = "<<add<End1;

Return 0;
```

Q52. Write a C++ program to access public data member inside main using object name

Ans. Program

```
#include <iostream>

Using namespace std;

Class Sample

{

        Public:

                Int x;

};

Int main ()

{

        Sample objSample;

        objSample . x = 20;

        cout <<"value of x: "<<objSample x<<end1;

        return 0;

}
```

Q53. Write a C++ program that demonstrates the definition of constructor and destructor inside class definition.

Ans. Program

```
#include <iostream>

Using namespace std;

Class Example

{

        Public:

                // default constructor

                Example (){cout<<"Constructor called"<<end1;

}

Int main ()

{

        // Object creation

        Example objE;

        ObjE. Display ()';

        Return 0;

}
```

Q54. Write a C++ program that demonstrates the definitions of constructor and destructor outside class definition

Ans. Program

```
#include <iostream>
```

```
Using namespace std;

Class example

{

Public:

//default constructor

Example ( );

//function to print message

Void  display ( );

//desctructor

~example ( );

};

//function definitions

Example ( )

{

Cout < <  "constructor called."< < end1;

}

Void example : display ( )

{

Cout  < <"display function called."< <end1;

}

Example : : ~example ( )

{

Cout < <"Destructor called. " < < end1;
```

```
}
Int main ( )
{
//object creation
Example objE;
objE.display ( );
return 0;
}
```

Q55. Write a C++ program that shows how global and local variable behave and to manipulate them.

Ans.

```
#include <iostream>
Using namespace std ;
//defining the global variable
Int a=10;
Int main ( )
{
//local variable
Int a=5;
Cout < < "local a: "<<a<<"Global a: "<<::a;
//Re.defining global variable by using : :
: : a=20;
```

Cout < < "\nlocal a : "< < a < < "Global a : "<<:: a;

Return 0;

}

Q56. Write a C++ program that read and print class, students details using two classes.

Ans

#include <iostream>

#include <conio.h>

Using namespace std ;

Class student

{

Private :

Char name [30];

Int roll no ;

Public :

Void getstudent ( )

{

Strcpy (name, "Fahad Shahzad");

rollNo=330;

}

Void printstudent ( )

{

Cout < <"Name: "name<<",Roll No: "<<roll no<<end1;

```
}
};
Class details
{
Private :
Char clsname [30];
Student std ; //object
Public:
Void getclassdetials ( )
{
Strcpy(clxsName, "M.Phill");
Std.getStudent( );
}
Void printClassDetails ( )
{
Cout < <"Class Name."< <clsName< <end1;
Std.printStudent ( );
}
};
Int main ( )
{
classDetails CD;
CD.getClassDetails ( );
```

```
CD.printClassDetails ( ):

Return 0:

}
```

Q57. Write a C++ program that prints the values of different data types using function overloading.

Ans.

```
#include <iostream.h>

#include <conio.h>

Class printData

{

Public :

Void print (int i)          //function 1

{

Cout < <"Printing int : "<< i < <"\n":

}

Void print (double f)       //function 2

Cout < <"Printing float : "< < f< < "\n":

}

Void print (char*c)

{

Cout  < <"Printing characters (string) : "<< c< <"\n";

}
```

```
};
Void main ( )
{
Clrscr ( ) :
PrintData pdobj:
Pdobj.print(5):          //called print ( ) to print integer
Pdobj.print(50.434):        //called print ( ) to print float
Pdobj.print("C++ Function Overloading"):                //called print ( ) to print string
Getch() :
}
```

Q58. Write a C++ program that stores price list of 5 items and prints the largest price as well as the sum of all prices using class

Ans

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
Class ITEM
{
Int itemcode[5]:
Float itprice [5]:
Public:
Void initialize(void):
```

```
Float largest(void);
Float sum(void);
Void displayitems(void);
};
Void ITEM: :initialize(void)
{
For (int i=0; i<5; i++)
{
Cout<<"Item No:"<<(i+1);
Cout<<"\nEnter item code:";
Cin>>itemcode[i];
Cout<<"Enter item price:";
Cin>>itprice[i];
Cout<<"\n";
}
}
Float ITEM: :largest (void)
{
Float larg=itprice[0];
For (int i=1; i<5; i++)
{
If (large<itprice[i])
{
```

```
Larg=itprice[i]:
}
}
Return larg:
}
Float ITEM: :sum(void)
{
Float sum=0;
For (int i=0: i<5: i++)
{
Sum=sum+itprice[i]:
}
Return sum:
}
Void ITEM: :displayitems(void)
{
Cout<<"\nCode\tPrice\n";
For (int i=0: i<5: i++)
{
Cout<<itemcode[i]<<"\t";
Cout<<itprice[i]<<"\n";
}
}
```

```
Void main ( )
{
Clrscr( ):
ITEM order;
Order.initialize( );
Float tot, big;
Int ch=0;
Do
{
Cout<<"\nMain Menu\n";
Cout<<"1. Display Largest Price\n";
Cout<<"2. Display Sum of Price\n";
Cout<<"3. Display Item List\n";
Cout<<"4. Exit\n";
Cout<<"Enter your choice(1-4):";
Cin>>ch;
Switch(ch)
{
Case 1:     big=order.largest( );
Cout<<"Largest Price=" <<big;
Break;
Case 2:     tot=order.sum( );
Cout<<"Sum of Prices=" <<tot;
```

```
Break;

Case 3:  order.displayitems( );

Break;

Case 4:  cout<<"Existing...press any key....";

Getch( );

Exit(1);

Default:cout<<"\nWrong choice...!!";

Break;

}

Cout<<"\n";

}

While (ch>=1 && ch<=4);

Getch( );

}
```

Q59. Write a C++ program that public data members of objects of a class can be accessed using the direct member access operator (.)

Ans. Program

```
#include <iostream.h>

#include <conio.h>

Class box

{

Public:
```

```
Double leng:      //length of a box
Double brea:      //breadth of a box
Double heig:      //height of a box
};
Void main ( )
{
Clrscr ( ):
Box box1obj:        //Declared box1obj of type box
Box box2obj:        //Declared box2obj of type box
Double volume=0.0      //Store the volume of a box here
//box 1 specification
Box1obj.heig=5.0;
Box1obj.leng=6.0;
Box1obj.brea=7.0;
//box 2 specification
Box2obj.heig=10.0;
Box2obj.leng=12.0;
Box2obj.brea=13.0;
//calculate volume of box 1
Volume=box1obj.heig*box1obj.leng*box1obj.brea;
Cout<<"Volume of Box1="<<volume<<"\n";
//calculate volume of box 2
Volume=box2obj.heig*box2obj.leng*box2obj.brea;
```

```
Cout<<"Volume of Box2="<<volume<<"\n";
Getch( );
```

Q60. Write a C++ program implementing inheritance between Employee (base class) and Manager (derived class)

Ans. Program

```
#include <iostream>
#include <bits/stdc++.h>
Using namespace std;
//Base class
Class employee
{
Public:
Int id_e;
;
//Sub class inheriting from Base Class (Parent)
Class Manager : Public Employee
{
Public:
Int id_m;
};
//main function
Int main ( )
```

```
{
Manager obj1:
//An object of class child has all data members
//and member functions of class parent
Obj1.id_m=5;
Obj1.id_e=10;
Cout<<"Child id is "<<obj1.id_m<<end1;
Cout<<"Parent id is"<<obj1.id_e<<end1;
Return 0;
}
```