```
cout<<Tester::var<<endl ;        // Line 3
}
```

A) 15 7 0
B) 7 15 0
C) 15 9 0
D) 9 15 0
E) Compilation error at line 1
F) Compilation error at line 2
G) Compilation error at line 3
H) None of the above.

Just write the letter(s) corresponding to your answer in the corresponding cell.
Answers that are not written in this table will NOT be marked nor considered.
Extra sheets of paper are NOT allowed to be attached to this exam.
You are NOT allowed to leave the room with the questions sheet.

Answer Sheet

Question #1 (True or False)		Question #2 (Choose all that apply)		
False	1.	C, F	15.	B
False	2.	A, B	16.	B
True	3.	B, C	17.	D
False	4.	B	18.	B
False	5.	A	19.	B
False	6.	B, C	20.	A, C
False	7.	C	21.	D
False	8.	C	22.	B
False	9.	D	23.	A, B
False	10.	A	24.	A, B
	11.	G	25.	E
	12.	B		
	13.	D		
	14.	B		

Q.1 True or False: (10 Marks)

1) The constructor of a class is responsible for removing the object from the memory (X)

2) Multilevel Inheritance is allowed as an object Oriented concept, but it is not allowed in C++. (✓)

3) Association is a special type of Composition (X)

4) The term "Encapsulation" refers to an object of a class that contains another object of another class inside it. (✓)

5) When inheriting from a Base class, the Derived class will inherit only the protected, and public members of the Base class. (X)

6) Static member variable can only be modified through static member functions. (X)

7) If a certain function is made friend for class A, then that function can access only the private members of class A. (X)

8) Composition is special type of Inheritance; composition is stronger relation between 2 objects than inheritance. (X)

9) The "this" pointer is an array of pointers to all the objects created from a class. (X)

10) When overloading a certain function, it is obligatory that we specify a different number of parameters for the new function. (X)

Q.2- Choose the correct answer(s) (Choose all that apply) (50 Marks)

Q.1 True or False: (10 Marks)

1) The constructor of a class is responsible for removing the object from the memory ( )

2) Multilevel Inheritance is allowed as an object Oriented concept, but it is not allowed in C++. ( )

3) Association is a special type of Composition ( )

4) The term "Encapsulation" refers to an object of a class that contains another object of another class inside it. ( )

5) When inheriting from a Base class, the Derived class will inherit only the protected, and public members of the Base class. ( )

6) Static member variable can only be modified through static member functions. ( )

7) If a certain function is made friend for class A, then that function can access only the private members of class A. ( )

8) Composition is special type of Inheritance; composition is stronger relation between 2 objects than inheritance. ( )

9) The "this" pointer is an array of pointers to all the objects created from a class. ( )

10) When overloading a certain function, it is obligatory that we specify a different number of parameters for the new function. ( )

Q.2- Choose the correct answer(s) (Choose all that apply) (50 Marks)

9) The "this" pointer is an [illegible]

10) When overloading a certain function, it is obligatory that we specify a different number of parameters for the new function. (X)

Q.2- Choose the correct answer(s) (Choose all that apply) (50 Marks)

1) Which of the following is true about an object member function?

A) It can be called using the name of the class.
B) It can access static variables of the class.
C) It has a "this" pointer as an implicit parameter passed to it.
D) It can access the instance variables.
E) It cannot be overloaded.
F) It can call other member functions from inside it.

2) Which of the following is true about the function prototype below?

```
void myFunc (int myDef=17, int myVar , int myNormalVar=5) ;
```

A) We should also give a default value to myVar.

B) We must only give a default parameter for myNormalVar and not the others.
C) The function is correct in that way.

this → a
*this.a

3) class Super

```
{
  protected:
        Super(int a)
          {
            this.a = a;     // Line 1
          }
//private:                  // Line 2
        int a;
};
class Sub: public Super
{
   public:
        Sub(int a):Super(a) { }
   public:
        Sub()               //Line 3
        {
          this.a= 5;        //Line 4
        }
};
```

Which steps will allow Sub to compile?

A) Class Sub compile successfully.
B) Comment Line 2
C) In Line 1 and 4 to, use (*this).a instead of this.a;
D) Change Line 1 and 4 to, this(a);
E) Change Line 3 to, Sub(): Super(5)
F) Change Line 3 to, Sub(): this(5)
G) All the above.

4) **What will be the output when you compile and run the following piece of code?**

```
class Test
   { int x;
    public:
       Test( ) { x = 0; }
       Test(int y) { x = y++; }
       Test(Test &r) { x = ++r.x; }
        void print( )    {cout<<x ; }
      };
   void main()
        {   Test t(1) ;
            t.print() ;
```

1

```
        Test x(t) ;
        x.print() ;
        t.print() ;
    }
```

A) 121
B) 122
C) 222
D) 233

5)
```
class Parent
    { public:
            int x;
            Parent(int m)  { x = m ; }
    };
class Child : protected Parent
    {
        public:
            int y;
            Child(int m, int n) : Parent(m) {  y = n ; }
    };
class GrandChild : public Child
    {   int z ;
        public:
            GrandChild(int a, int b, int c) : Child(a,b) {  z = c ; }
    };
void main( )
    {  GrandChild obj(3,5,7);
       cout<<"Value of x is: "<<obj.x <<endl ;          //Line 1
       cout<<"Value of y is: "<<obj.y <<endl ;          //Line 2
       cout<<"Value of z is: "<<obj.z <<endl ;          //Line 3
    }
```

A) Compiler Error at Line 1
B) Compiler Error at Line 2
C) Compiler Error at Line 3
D) The code compiles successfully.

6)What will be the output when you compile and run the following piece of c

```
class Parent
 {
    int x;
    Parent(int m){ x = m ; }
 };
class Child : public Parent
 { public:
```

A) Compilation Error at Line 1, an object member function cannot access a static member
B) Compilation Error at Line 2, constructor should initialize static member (z=0;)
C) Compilation Error at Line 3
D) Compilation Error at Line 4
E) The code compiles successfully.

9) What will be the output when you compile and run the following piece of code?

```
class Parent
  {
    protected:
          int x;
    public:
          Parent(int m)
                { x = m ; }
    friend void display( );
  };
class Child : public Parent
  {
    private:
          int y;
    public:
        Child(int m, int n) : Parent(m)
        {  y = n ;  }
  };
void display ()
        {
          Child c(3,4);
          cout <<"x="<<c.x<<"y="<<c.y; // Line 1
        }

void main ()
        {
          display();
        }
```

A) Compilation Error at Line 1, Child::x is inaccessible
B) Compilation Error at Line 1, Child::y is inaccessible
C) A and B
D) The code compiles successfully.

```
        int y;
        Child(int m, int n) : Parent(m) //Line 1
        { y = n ; }
};

void main( )
{
  Child obj(3,5,7);          //Line 2
  cout<<"Value of x is: "<<obj.x <<endl ; //Line 3
  cout<<"Value of y is: "<<obj.y <<endl ; //Line 4
}
```

A) Compiler Error at Line 1
B) Compiler Error at Line 2
C) Compiler Error at Line 3
D) The code compiles successfully.

7) In order to turn a class into an abstract class, which of the following do we need to do?

A) Write the abstract keyword before the name of the class.
B) Make the class a pure virtual class.
C) Write one or more pure virtual functions inside the class.
D) A and C.
E) None of the above

8) What will be the output when you compile and run the following piece of code?

```
class Parent
      { int y;
        static int z;
        public:
              Parent( )
                  {
                    z=0;  // Line1
                  }
              Parent (int a=5)  //Line 2
                  {
                    y=a;
                  }
      };
void main( )
      {
         Parent d(4);    //Line 3
         Parent m;       //Line 4
      }
```

```
        Test x(t) ;
        x.print() ;
        t.print() ;
    }
```

A) 121
B) 122
C) 222
D) 233

5)
```
class Parent
    { public:
            int x;
            Parent(int m)  { x = m ; }
    };
class Child : protected Parent
    {
        public:
            int y;
            Child(int m, int n) : Parent(m) {  y = n ;  }
    };
 class GrandChild : public Child
    {   int z ;
        public:
          GrandChild(int a, int b, int c) : Child(a,b) {  z = c ; }
    };
void main( )
    {  GrandChild obj(3,5,7);
       cout<<"Value of x is: "<<obj.x <<endl ;          //Line 1
       cout<<"Value of y is: "<<obj.y <<endl ;          //Line 2
       cout<<"Value of z is: "<<obj.z <<endl ;          //Line 3
    }
```

A) Compiler Error at Line 1
B) Compiler Error at Line 2
C) Compiler Error at Line 3
D) The code compiles successfully.

6) What will be the output when you compile and run the following piece of code

```
class Parent
 {
    int x;
    Parent(int m){ x = m ; }
 };
 class Child : public Parent
 { public:
```

10) What will be the output when you compile and run the following piece of code?

```
class Nice
    {  int a ;
        public:
            Nice( ) { a = 0 ; }
            Nice(Nice & myN)
                {  this →a =  myN.a ;
                   cout<<"I am the copy constructor\n" ;
                 }
            void setA(int m)  { a = m; }
             int getA()   { return a ; }
    };
void show(Nice &obj)
{   cout<<"I am the show function, value is: " << obj.getA() ; }
void main()
{  Nice n1;
   n1.setA(15) ;
   show(n1) ;
}
```

A) I am the show function, value is: 15.
B) I am the show function, value is: 15.
   I am the copy constructor.
C) I am the copy constructor.
   I am the show function, value is: 15.
D) I am the copy constructor.

11) What will be the output when you compile and run the following piece of code?

```
class A
{
   int x;
   protected :   int y ;
   public   :
            A(int x1=5,int y1=3)  { x=x1;y=y1; }
            void M1( ) {cout <<"\n This is M1() in class A:Base class";}
            void M3( ) {cout <<"\n This is M3() in class A:Base class";}
```

```
};
class B:private A
      {
         int w;
         protected:   int v;
         public  :
                  B(int v1=3, int w1=9) { v=v1;w=w1; }
                  void M3()
                   { M1( );  //Line 1
                     y++;    //Line2
                   }
      };
class C: public B
      {   public:
          void  M4( )
                { M1( );  //Line 3
                  y++;    //Line 4
                }
      };
void main()
      { B b1;
        b1.M3( );     //Line 5
        b1.M1( );     //Line 6
      }
```

A) Compilation Error at Line 1.
B) Compilation Error at Line 2.
C) Compilation Error at Line 3.
D) Compilation Error at Line 4.
E) Compilation Error at Line 5.
F) Compilation Error at Line 6.
G) The code compiles successfully.

12) Which of the following most closely describes the process of overriding?

A) A class with the same name replaces the functionality of a class defined earlier in the hierarchy.
B) A function with the same name replaces the functionality of a function defined earlier in the inheritance hierarchy.
C) A function with the same name but different parameters gives multiple uses for the same function name.
D) Making a class abstract so that no objects can be declared from it.

13) **"A plane is a machine that has a motor and has wings".**
"A refrigerator is a machine that has a motor and has shelves".
Which of the following best describes the previous statements as a set of classes?

A) 1 class: A machine class that has an attribute for the type of machine.
B) 2 classes: A plane class that has two attributes, and a refrigerator class that also has two attributes.
C) 3 classes: A machine class that has one attribute: motor. A plane class that inherits from the machine class. And a refrigerator class that inherits from the plane class.
D) 3 classes: A machine class that has one attribute: motor. A plane class that inherits from the machine class. And a refrigerator class that also inherits from the machine class.

14) **If we did not specify a constructor to the class, then :**
A) we won't be able to create object of class
B) we won't be able to create object of class, and compiler will give compilation error
C) we won't be able to create object of class, and compiler will give warning
D) it will generate run-time error
E) None of the above

N *n;

15) **Assume you have a class M that contains a pointer to an object of class N. Assume that we declare an object of M in the main( ) function. When will the body of the constructor of class N be executed?**

A) When any member function of the class M is called.
B) After the body of the constructor of class M is executed.
C) Before the body of the constructor of class M is executed.
D) None of the above.

16) **What will be the output when you compile and run the following piece of code?**

```
class Parent
    {
     protected:
        int myVar;
     public:
        Parent(int x) { myVar=x; }
        void powerTwo(){ cout<<myVar*myVar; }
        virtual void powerThree() { cout <<myVar*myVar*myVar; }
    };
class Child:public Parent
    {
     protected:
         int myData;
```

```
{ public:
        Derived() { cout<<"Hello     " ; }
};
void main()
{  Base myBase ;
   Derived myDerived ;
}
```

A) Welcome Hello
B) Hello Welcome
C) Welcome Hello Welcome
D) Welcome Welcome Hello

22) What does the following piece of code do?

```
void main( )
    {
     float *ptr ;
    ptr = new float[15];
    }
```

A) Allocate space for a float variable that is not initialized
B) Allocate space for an array of 15 float elements that are not initialized
C) Allocate space for an array of 15 float elements that is initialized by the value 0
D) Allocate space for an array of 15 float elements where all the elements are initialized by the value 15
E) Compiler Error.

23) Which of the following statements are true about constructor?

A) A constructor can be overloaded.
B) A constructor is a special member function with the same name of the class.
C) A constructor can return a primitive or an object reference.
D) All the above

24) What will be the output when you compile and run the following piece of code?

```
class GrandFather
        { public:
            virtual void displayStuff() = 0 ;
            virtual void sayThings() = 0 ;
        };
class Parent : public GrandFather
```

}

what shall we add to the above class Stack to declare another object s2 from class Stack where s2 is declared in terms of s1; Stack s2(s1);

A) We must specify overload of assignment operator for class Stack.
B) We must define a copy constructor to class Stack
C) A and B
D) This situation cannot be achieved in C++, however, it has been solved in other programming languages.

19) class Point

```
{
 float x, y;
Point (float a, float b) { x=a; y=b;}
Point () { x=0; y=0; }
}
```

To write copy constructor to class point, what would be its signature?

A) Point (Point)
B) Point (Point &)
C) Point & Point (Point &).
D) Point & Point (Point)
E) void Point (Point)
F) void Point (Point &)
G) None of the above

20) Assume you have a member function with the following prototype?

```
void myFunc(int x) ;
```

Which of the following are valid ways to overload it?

A) void myFunc(char ch) ;
B) int myFunc(int x) ;
C) void myFunc(char c1, char c2) ;
D) float myFunc(int x, int y) ;

21)What will be the output when you compile and run the following piece of code?

```
class Base
    {  public:
            Base() { cout<<"Welcome     " ; }

    };
```

```
        public:
            Child(int a, int b) : Parent(a) {myData= b;}
            void powerTwo() { cout <<myData*myData; }
            void powerThree() { cout<<myData*myData*myData; }
        };
    void main()
    {
      Child myCh(2,3);
      Parent *myPtr;
     myPtr = &myCh;
     myPtr->powerTwo(); //Line1
     myPtr->powerThree(); //Line 2
}
```

A) 4 8
B) 4 27
C) 9 27
D) 9 8
E) Compilation Error at Line1
F) Compilation Error at Line2

17) In order for the following piece of code to compile successfully, what are the constructors that are expected to exist in the Base class?

```
class Child : public Base
        {
            public:
                Child(int x)  { }
                Child(int x, int y) : Base(x,y) { }

        };
```

A) Base( ) and Base(int , int).
B) Base( ) and Base(int).
C) Base(int) and Base(int , int).
D) Base(int , int).

18)

```
class Stack
  {
    int tos,size;
    int * st;
    Stack( int s=5) { tos=0;size=s; st=new int[size];}
    ~ Stack( ){ delete []st;}
  };
void main()
{
```

```
        { public:
              void displayStuff( ) { cout<<"Parent's Stuff" ; }
        };
class Child : public Parent
        { public:
              void sayThings( ) { cout<<"Child's Things" ; }
        };
class GrandChild : public Child
        { public:
             void displayStuff( ) { cout<<"GrandChild's Stuff" ; }
             void sayThings( )    { cout<<"GrandChild's Things" ; }
        };
void main( )
        { GrandFather myGF ; // Line 1
          Parent myP ;       // Line 2
          Child myCh ;       // Line 3
          GrandChild myGC ;  // Line 4
          GrandFather * ptr ; // Line 5
        }
```

A) Compiler Error at Line 1.
B) Compiler Error at Line 2.
C) Compiler Error at Line 3.
D) Compiler Error at Line 4.
E) Compiler Error at Line 5.
F) The code compiles successfully.

25) What will be the output when you compile and run the following piece of code?

```
class Tester
 {  public:
             int x ;
            static int var ;
             Tester(int a) {  x = a ; }
            static void myFunction(int a)
               {
                  Tester obj(9) ;      // Line 1
                  obj.x=a ;            // Line 2
                  cout<<obj.x;
               }
 };
 int Tester::var=0;
void main()
{
 Tester myT(7) ;
 Tester::myFunction(15);
 cout<<myT.x ;
```