# Intake 42 OOP Exam
# Duration : 90 Min. (3)

...

* Required

1. Full Name ( in English ) *

Enter your answer

2. Track *

- [x] BI
- [ ] Mobile Cross
- [ ] Open Source

3. A protected member of a class can be directly accessed by its name inside another class if and only if that other class is a child of that class.
(1 Point)

- [ ] True
- [ ] False

4. Which of the following statements are true about constructor?
(3 Points)

- [ ] A constructor can not be overloaded.
- [ ] A constructor is a special member function with the same name of the class.
- [ ] A constructor can return void
- [ ] All the above

5. What does the following piece of code do?

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

(3 Points)

- ( ) Allocate space for a float variable that is initialized to 15
- ( ) Allocate space for an array of 15 float elements that are not initialized
- ( ) Allocate space for an array of 15 float elements that is initialized by the value 0
- ( ) Allocate space for an array of 15 float elements where all the elements are initialized by the value 15
- ( ) Compiler Error.

6. When inheriting from a Parent class, the Child class will inherit the private members of the parent class
(1 Point)

- ( ) True

- ○ False

7. Which of the following is true about the function prototype below?
   void add (int myDef, int myVar=6 , int myNormalVar=5) ;
(3 Points)

- ○ We should also give a default value to myDef.
- ○ We must only give a default parameter for myNormalVar and not the others.
- ○ The function is correct in that way.

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)   //Line 2
                        {
                           y=a;
                         }
         };
  void main( )
          {
             Parent d(4);     //Line 3
             Parent m;        //Line 4
           }
```

(3 Points)

- ○ Compilation Error at Line 1, an object member function cannot access a static member

- Compilation Error at Line 2, constructor should initialize static member (z=0;)
- Compilation Error at Line 3
- Compilation Error at Line 4.
- The code compiles successfully.

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

- True
- False

10. When overloading a certain function, the only way is to specify a different number of parameters for the new function.
(1 Point)

- True
- False

11. The relation between the car object and the driver object is represented as .....relation
(1 Point)

- Composition
- Aggregation
- Association

Inheritance

12. In order to turn a class into an abstract class, which of the following do we need to do? (3 Points)

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

13. 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();
 }
```

(3 Points)

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

14. Which of the following is true about an object member function?
(4 Points)

- ☐ It can be called using the name of the class.
- ☐ It can access static variables of the class.
- ☐ It has a "this" pointer as an implicit parameter passed to it.
- ☐ It can access the instance variables.
- ☐ It cannot be overloaded.
- ☐ It can call other member functions from inside it.

15. If we did not specify a constructor to the class, then :
(3 Points)

- ○ we won't be able to create object of class
- ○ we won't be able to create object of class, and compiler will give compilation error

we won't be able to create object of class, and compiler will give compilation error

we won't be able to create object of class, and compiler will give warning

it will generate run-time error

None of the above

16. 
```
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
}
```

(3 Points)

- [ ] Compiler Error at Line 1
- [ ] Compiler Error at Line 2
- [ ] Compiler Error at Line 3
- [ ] The code compiles successfully.

17. We can overload Destructor in the class

(1 Point)

- ( ) True
- ( ) False

18. 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, int y) : Base(x,y) { }
        };
```

(3 Points)

- ( ) Base( ) and Base(int , int).
- ( ) Base( ) and Base(int).
- ( ) Base(int) and Base(int , int).
- ( ) Base(int , int)

Base(int , int);

19. the relation between the Lecture object and the Instructor object is represented as .....relation
(1 Point)

- Composition
- Aggregation
- Association
- Inheritance

20. what is the output?

```
class Card
{
   int a ;
public:
   Card( )
   {
      a = 0 ;
       cout<<"I am the default constructor " ;
   }
   Card(Card & myN)
   {
      this -> a = myN.a ;
      cout<<"I am the copy constructor " ;
   }
   void setA(int m)
   {
      a = m;
   }
   int getA()
   {
      return a ;
   }
```

```
};
void show(Card obj)
{
   cout<<"I am the show function, value is: " << obj.getA() ;
}
void main()
{
   Card n1;
   n1.setA(15) ;
   show(n1) ;
}
```

(3 Points)

- I am the default constructor .I am the show function, value is: 15.
- I am the default constructor I am the show function, value is: 15. I am the copy constructor.
- I am the default constructor I am the copy constructor. I am the show function, value is: 15.
- I am the copy constructor. I am the default constructor

21. "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?
(3 Points)

- 1 class: A machine class that has an attribute for the type of machine.
- 2 classes: A plane class that has two attributes, and a refrigerator class that also has two attributes.
- 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.
- 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.

22. The term " Composition " refers to an object of a class that contains a pointer to another object

(1 Point)

- ○ True
- ○ False

23. Assume you have a member function with the following prototype?
    void myFunc(int x) ;
    Which of the following are valid ways to overload it?
    (3 Points)

- ☐ void myFunc(char ch) ;
- ☐ int myFunc(int x) ;
- ☐ void myFunc(char c1, char c2) ;

24. Assume you have a class M that contains 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?
    (3 Points)

- ○ When any member function of the class M is called.
- ○ After the body of the constructor of class M is executed.
- ○ Before the body of the constructor of class M is executed.
- ○ None of the above.

25.
```
class Point{
   int x,y;
public:
   void setX(int _x){x=_x;}
   void setY(int  y){y= y;}
```

```
    void setXY(int _x,int _y){x=_x;y=_y;}
    int getX(){return x;}
    int getY(){return y;}
    Point(int _x,int _y){
        x=_x;
        y=_y;
        cout<<" Point Constructor";
    }
    Point(int xy){
        x=y=xy;
        cout<<" Point Constructor";
    }
};
class Rectangle{
    Point ul,lr;
public:
    void setUL(int _x,int _y){
        ul.setXY(_x,_y);
    }
    void setLR(int _x,int _y){
        lr.setXY(_x,_y);
    }

    Rectangle(int x1,int y1,int x2,int y2):ul(x1,y1)
    {
        cout<<"\n  Rectangle Constructor";
    }

};
```

**what is the output when you create object of rectangle. Rectangle r(5,6,7,8);**
(3 Points)

- ◯ Compilation Error
- ◯ Rectangle Constructor  Point Constructor
- ◯ Rectangle Constructor
- ◯ Point Constructor Rectangle Constructor

26. What will be the output when you compile and run the following piece of code?