Object –Oriented Programming Final Exam.

Q1: True or False.

1. Any static member can access all static and instance members. (F)
2. You can have more than a constructor in the same class. (T)
3. A destructor is a member function that removes object from memory. (F)
4. An abstract class is a class that contains any virtual functions. (F)
6. A constructor is always declared void because it doesn't return a value. (F)
7. Aggregation and association are types of relationships between a base class and derived classes. (F)
8. The embedded object is an object declared with default constructor. (F)
9. You cannot overload the scope operator (::). (T)
10. You can define a copy constructor in class. (T)

Q2: Select the correct answer(s).

1. Object-Oriented is used because: (2 answers)
   - The code is smaller.
   - The program is faster.
   - The code is reusable.
   - The code is easy to maintain.

2. Which of the following refers to "Hiding data inside class methods" ?
   - Abstraction
   - Encapsulation
   - Polymorphism
   - Inheritance

3. We use the inheritance for:
   - Increasing more member functions only to the base class.
   - Extending base class functionality.
   - Including the base class.

4. The following are examples of polymorphism: (2 answers)
   - Operator overloading
   - Function overriding
   - Friend functions
   - Constructor overloading.

5.
   ```
   class XX
   {
           int i;
           char c;
           void seti (int r) { }
           char getc ( ) { return c; }
   };
   void main( )
   {
           XX x;
           x.seti(8);
   }
   ```

   On running, the previous code:

   - compiles with no errors
   - the compiler generates an error because the class XX cannot be instantiated because it has no public constructor.
   - The linker generates an error because the class XX cannot be instantiated because it has no public constructor.
   - [✓ The compiler generates an error because the method "seti" is non accessible through object x.]

6.
   ```
   void printNumber( )
   {
           Int num=5;
           for (i=0 ; i<10 ; i++)
           { cout<<num; }
           int i;
           char c;
   }
   ```

   In C++, on calling the previous function:

   - An error occurs because i & c cannot be declared after the loop (invalid declaration).
   - [✓ An error occurs because the var i is not visible in the for loop (undefined symbol i)]
   - No error occurs.

7. The constructor of the base class called automatically:

   - When any member function of the derived class called.
   - After the constructor of the derived class started.
   - [✓ Before the constructor of the derived class started.]
   - We do not know when exactly it will be started.

8.
```
class X
{
        int k;
        public:
                void setK(int k) {……..}
};
```
Which can best be written in the space?
- [✓ this->k=k;]
- k=k;
- this.k=k;

9.
```
class Person
{
        public:
                virtual void talk( )==0;
};
class Adult : public Person
{
        public:
                void talk() { cout<<”Hello”; }
};
class Female : public Person
{
        public:
                void talk( ) { cout<<”Hi”; }
};
void main( )
{
        Person p;
        Adult a;
        Female f;
}
```
The previous code:
- Compiles successfully.
- [✓ Produces an error.]

10.
```
class Child : public Base
{
        public:
                Child(int x) { }
                Child(int x,int y) : Base(x,y)
                { }
};
```

What are the expected constructors that must exist in the base class?

- Base( ), Base(int,int)
- Base( ), Base(int)
- Base(int), Base(int,int)
- Base(int,int)

11. The class that contains a pure virtual function is called:

- Base class.
- Derived class.
- Abstract class.
- Embedded class.

12. #include <iostream.h>

```
class XX
{
        int i;
        char c;

        public:
                void seti(int r) { }
                void getc( ) { return c; }
                void printHi( ) { cout<<"Hi"; }
};
void printHi( ) { cout<<"Hello"; }
void main( )
{
        XX x;
        x.printHi( );
}
```

The previous code prints:

- Hello.
- Hi.
- Hi then Hello
- None of the above, it generates an error.

13. class Student

```
{
        char name[20];
        public:
                Student( ) { strcpy(name, "Anonymous"); cout<<name; }
                Student(char* nm) { strcpy(name,nm); cout<<name; }
                ~Student( ) { }
};
void main( )
{
        Student s1(Mona);           ……1
        Student *s2;                ……2
        S2 = new Student("Ali");    ……3
        delete s1;                  ……4
        delete s2;                  ……5
}
```

In the previous code:

- It compiles successfully.
- Line 4 generates an error.
- Line 5 generates an error.
- Both lines 4,5 generate errors.

14. class Test

```
{
        public:
                void print(int i)
                { cout<<"int version"; }
                void print(char *c)
                { cout<<"char version"; }
};

void main( )
{
        Test t;
        char ch='p';
        t.print(ch);
}
```

Which of the statements below is true?

- Line 4 will not compile, because void methods cannot be overloaded.
- Line 7 will not compile, because there is no version of print that takes a char.
- The code will compile and produce the following output: int version.
- The code will compile and produce the following output: char version.

15. class Base

```
{
        public:
                int x;
};

class Child : private Base
{
        public:
                int y;
};

void main( )
{
        Child c;
        c.x=10;
        cout<<"x="<<x;
}
```

What happens in program?

- Compiler error because x is not declared in Child class.
- Compiler error because of the Child inheritance from private class.
- x=10
- Runtime error.

Q3: Design Question.

Design a set of classes to calculate the volume of the following shapes:

Cylinder	($\pi r^2 l$), where r: radius, l: length.
Cube	(l^3), where l: length.
Box	(w*h*l), where w: width, h: height, l: length.

Answers:

Q1: 1. F
2. T
3. F
4. F
5. T
6. F
7. F

8. F
9. T
10. T

Q2:
1. c, d
2. b
3. b
4. a
5. d
6. b
7. c
8. a
9. b
10. a
11. c
12. d
13. b
14. c
15. b

Q3:
```
class Shape
{
        protected:
                int length;
        public:
                Shape( ) { length=0; }
                Shape( int l ) { length=l; }
                virtual void calc_volume( )=0;
};
class Cylinder : private Shape
{
        private:
                int radius;
        public:
                Cylinder( ) : Shape( )
                { radius = 0; }
                Cylinder (int l,int r) : Shape(l)
                { radius = r; }
                void calc_volume( )
                {
                        float result;
                        result = (22/7)*radius*radius*length;
                        cout<<"Cylinder volume= "<<result<<endl;
```

```
                    }
};
class Box : public Shape
{
          protected:
                    int width,height;
          public:
                    Box( ) : Shape( )
                    { width = height = 0; }
                    Box(int l, int w, int h) : Shape(l)
                    {
                              width = w;
                              height = h;
                    }
                    void calc_volume( )
                    {
                              int result;
                              result = width * height * length;
                              cout<<"Box volume= "<<result<<endl;
                    }
};
class Cube : private Box
{
          public
                    Cube( ) : Box( ) { }
                    Cube(int w) : Box(w,w,w)
                    { }
};
```