National University of Sciences and Technology
College Of E&ME
1^{st} Sessional, Fall 2013
DE-34 MTS

Subject Code: EC 204 | Subject: Data Structures & OOP
Date: October 22, 2013 | Timing: 9:50am-10:50am
Total Marks: 50 | Time: 1 hour

Name : _____ Section: _____

Registration Number: _____

Instructor: Ahsan Ijaz

---

1. Fill in the following blanks:
   (a) (1 point) Constructors are called when objects are _____.

   (b) (1 point) Destructors are used to _____.

   (c) (1 point) Every Object can access its own members using _____ pointer.

   (d) (2 points) In single inheritance, constructor of _____ class is called before the call of _____ class constructor.

2. Indicate as true or false. In case of false, suggest correction:
   (a) (1 point) Constructor can be treated as a public member function and can therefore have any return type.
   ◯ True ◯ False

   (b) (1 point) In Public inheritance, protected members can be accessed in derived class.
   ◯ True ◯ False

   (c) (1 point) Two global functions, with the same name, same arguments and same return type can be called in main by defining them in different places.
   ◯ True ◯ False

   (d) (1 point) Friend functions cannot access protected members.
   ◯ True ◯ False

   (e) (1 point) Destructors need to be explicitly called before going out of scope.
   ◯ True ◯ False

3. (15 points) In the table below, cross out the wrongly placed "yes". As an example, private members in public inheritance cannot be accessed in derived class and is hence crossed out in the table.

Accessibility from	Inheritance Type	Public	Protected	Private
Derived Class	Public	Yes	Yes	~~Yes~~
	Private	Yes	Yes	Yes
	Protected	Yes	Yes	Yes
Main Function	Public	Yes	Yes	Yes
	Private	Yes	Yes	Yes
	Protected	Yes	Yes	Yes

Table 1: Member access in Inheritance

 Please go on to the next page...

4. Refer to the following code snippet for answering the following question parts:

```
1  #include <iostream>
2  using namespace std;
3  class Furniture
4  {
5     public:
6        Furniture( int len );
7        Furniture( const Furniture &obj);
8        Furniture();
9        ~Furniture();
10   private:
11       int *ptr;
12 };
13
14 Furniture::Furniture(int len)
15 {
16     ptr=new int;
17     *ptr = len;
18 }
19
20 Furniture::Furniture()
21 {
22 }
23
24 Furniture::Furniture(const Furniture &obj)
25 {
26     ptr = new int;
27    *ptr = *obj.ptr;
28 }
29 Furniture::~Furniture(void)
30 {
31     delete ptr;
32 }
```

(a) (3 points) Identify constructor, copy constructor, constructor overloading, destructor and member access specifiers from the given code snippet (just write line number, e.g. Class Definition: Line# 3)

(b) (5 points) Error would occur in case of object instantiation (e.g. Furniture bed;) using the above code, identify the source and type of error.

(c) (2 points) What is the advantage of using & operator in the copy constructor. Talk in terms of memory allocation.

5. Write a code that:
   (a) (2 points) Has a class named **CoolingDevices** with public members as **Type** and **price**.
   (b) (3 points) Has a class inherited from **CoolingDevices** named **AirConditioner** with public members **price**, **PowerConsumption**, **CoolingCapacity** and a private member **thermostat**.
   (c) (5 points) Overload operator **"+"** so that by adding two objects of type **AirConditioner**, there public members **PowerConsumption** and **CoolingCapacity** are added together.
   (d) (5 points) Both the classes have the same member **price**. Briefly discuss scope of the variable price and how either one of it can be accessed in the derived class.

This exam has 5 questions, for a total of 50 points.

Question:	1	2	3	4	5	Total
Points:	5	5	15	10	15	50
Score:						

End of exam.