

Computer Programming :: Midterm Exam # 2

April 11, 2013

Max Points: 35

Time: 90 mins

Answer the following questions briefly and to the point.

1. Write three distinct situations in which copy constructor of a class is called. [3]
2. Consider the following class hierarchy and the corresponding main function. What is the output of this program? [5]

Class Hierarchy	Driver program
<pre>1 class D { 2 public: 3 D() { cout << "D ctor" << endl; } 4 D(D&) { cout << "D copy ctor" << endl;} 5 ~D() { cout << "D dtor" << endl; } 6 }; 7 8 class A { 9 public: 10 A() { cout << "A ctor" << endl; } 11 ~A() { cout << "A dtor" << endl; } 12 }; 13 14 class B : public A { 15 public: 16 B() { cout << "B ctor" << endl; } 17 ~B() { cout << "B dtor" << endl; } 18 void test(D d) { A a; } 19 }; 20</pre>	<pre>1 B globalB; 2 int main() 3 { 4 A a; 5 D d; 6 D d2 = d; 7 d = d2; 8 globalB.test(d); 9 return 0; 10 }</pre>

3. Given the classes above, what will be the output if *only* the following statement is written in main. Explain your answer. [2]

D * obj;

4. Is the piece of code below correct? If yes, what is the output of the function Bar. If no, why not? [3]

<pre>1 class Foo { 2 int x; 3 static int count; 4 public: 5 static int Bar(int i) { 6 return x*i*count + x*i + 1; 7 } 8 }; </pre>

5. Given the code in question 4, write code to initialize the data member count to 10. Also write the code to call the function Bar from main. [2]

6. Following is a C++ class representing a mathematical fraction, where n is the numerator and d is the denominator. Implement the post-decrement and pre-decrement operators for this class. [3]

```
1 class Fraction {
2     int n,d;
3 };
```

7. What is an initializer list? Describe two of its uses by giving examples. [3]
 8. Under which access specifier are friend functions and classes defined? [1]
 9. What is the difference between the keywords `struct` and `class`. [1]
 10. The following code on the left side lists a driver for a class `IntegerSet`. When executed, the code prints the output given on the right side.

<pre>1 int capacity = 10; 2 IntegerSet set1(capacity); 3 cout << "set1 = " << set1 << endl; 4 5 set1 += 2; // Add an element to the set 6 set1 += 5; 7 cout << "set1 = " << set1 << endl; 8 9 int arr[] = {1,2,3}; 10 IntegerSet set2(arr,3); 11 cout << "set2 = " << set2 << endl; 12 13 set2 += set1; // Union operation 14 cout << "set2 = " << set2 << endl; 15 16 IntegerSet set3 = set2; 17 if (set2 == set3) 18 cout << set2 << " == " << set3 << endl; 19 else 20 cout << set2 << " != " << set3 << endl; 21 22 set2 = set3 - set1; // Set difference 23 24 if (set2 == set3) 25 cout << set2 << " == " << set3 << endl; 26 else 27 cout << set2 << " != " << set3 << endl; 28 29 set2 = 2 + set2; // Add an element to set 30 cout << "set2 = " << set2 << endl;</pre>	<pre>set1 = [] set1 = [2 5] set2 = [1 2 3] set2 = [1 2 3 5] [1 2 3 5] == [1 2 3 5] [1 3] != [1 2 3 5] set2 = [1 2 5]</pre>
---	---

- Define the class `IntegerSet`. [2]
- Write declaration of all the functions which will allow the code above to run without any errors. You need to provide the interface only, no implementation is necessary. [6]
- Provide implementation of the functions corresponding to the operations at line 22. [4]