COEN 79

Object-Oriented Programming and Advanced Data Structures

Assignment #2 -

Name: Date:

Points per question: 1, unless noted

1. What is *encapsulation in OOP*? Give an example

2. (2 pts) What is the time complexity of `fun()`. Please show your proof.

```
int fun(int n)
{
  int count = 0;
  for (int i = n; i > 0; i /= 2)
     for (int j = 0; j < i; j++)
        count += 1;
  return count;
}
```

3. Give a concise formula that gives *the approximate number of digits in a positive integer*. The integer is written in base 10.

4. What are the differences between references and pointers?

5. What are the three ways we can use items defined in a namespace. Include examples in your answer.

6. (2 pts) Discuss about the output of the following code. How the result will change if we replace struct with class?

```
1.  struct Test {
2.      int x;
3.  };
4.
5.  int main() {
6.      Test t;
7.      t.x = 20;
8.      cout<t.x<endl;
9.      return 0;
10. }
```

7. (2 pts) A The header of the `point` class is as follows:

```
1.  class point
2.  {
3.  public:
4.      // CONSTRUCTOR
5.      point (double initial_x = 0.0, double initial_y = 0.0);
6.
7.      // MODIFICATION MEMBER FUNCTIONS
8.      void set_x (double& value);
9.      void set_y (double& value);
10.
11.     // CONST MEMBER FUNCTIONS
12.     point operator+ (double& in) const;
13.
14. private:
15.     double x; // x coordinate of this point
16.     double y; // y coordinate of this point
17.
18. };
```

- Which line of the following code results in an error? Explain why.
- What's the solution?

```
1.  main() {
2.      point myPoint1, myPoint2, myPoint3;
3.      double shift = 8.5;
4.      myPoint1 = shift + myPoint2;
5.      myPoint3 = myPoint1.operator+ (shift);
6.      myPoint1 = myPoint1 + shift;
7.  }
```

8. (2 pts) What is the output of this code? Discuss your answer.

```
1.  #include < iostream >
2.  using namespace std;
3.
4.  class CMyClass {
5.      public:
6.          static int m_i;
7.  };
8.
9.  int CMyClass::m_i = 0;
10.
11. CMyClass myObject1;
12. CMyClass myObject2;
13. CMyClass myObject3;
14.
15. int main() {
16.     CMyClass::m_i = 2;
17.     myObject1.m_i = 1;
18.
19.     cout << myObject1.m_i << endl;
20.     cout << myObject2.m_i << endl;
21.
22.     myObject2.m_i = 3;
23.     myObject3.m_i = 4;
24.
25.     cout << myObject1.m_i << endl;
26.     cout << myObject2.m_i << endl;
27. }
```