CS 215 – Study Questions 8
Class Syntax and Elements
Inheritance and Polymorphism

1. The file Circle.h contains the following definition of a class called Circle:

```cpp
#ifndef CIRCLE_H
#define CIRCLE_H

#include <iostream>
using namespace std;

class Circle
{
    public:
        Circle();
        Circle(float L);
        string GetType () const;
        void SetLength (float L);
        float GetLength () const;
        float GetArea () const;
        float GetPerimeter () const;
        friend ostream & operator <<
                        (ostream & outs, const Circle & C);
    private:
        float length;
};
#endif
```

a. What is the purpose of the lines
   ```cpp
   #ifndef CIRCLE_H
   #define CIRCLE_H
   ```
   and
   ```cpp
   #endif
   ```

b. Circle(); is the prototype for what function? Write an application statement that calls this function.

c. Circle(float L); is the prototype for what function? Write an application statement that calls this function.

d. Why are the functions GetType, GetLength, GetArea and GetPerimeter indicated as being "const"? Why is SetLength not "const"?
2. In the file Circle.cpp, the output operator is implemented as:

```cpp
ostream & operator << (ostream & outs, const Circle & C) {
    outs << "The area of your " << C.GetType();
    outs << " is " << C.GetArea();
    outs << "; its perimeter is: " << C.GetPerimeter();
    return outs;
}
```

d. Why does this function use the "C." when it calls the GetType, GetArea, and GetPerimeter functions? Would the statement:
   ```cpp
   outs << " is " << PI / 4 * length * length;
   ```
   work equally as well?

3. Write the prototype for an operator function that will determine if two circles are equal (==).

4. Write the implementation of your == operator function.

5. Write a segment of an application program that calls your == operator function.

6. If both the Circle class and the Square class have an == operator function, which one will be used by the following code segment?
   ```cpp
   Circle C1 (10);
   Square S1 (10);
   if (C1 == S1)
       cout << "C1 and S1 are the same" << endl;
   ```

7. In the application program, mySquares is declared as:
   ```cpp
   vector <Square *> mySquares;
   ```
   and the code
   ```cpp
   for (int i = 0; i < mySquares.size(); i++)
       cout << *mySquares[i] << endl;
   ```
   is used to write out a list of the squares.

   a. What is the purpose of the * in the declaration?

   b. What is the purpose of the * in the output statement?

   c. Write a look to write out a list of the circles stored in someCircles:
      ```cpp
      vector <Circle> someCircles;
      ```

8. What is the function of the "new" verb in the statement:
   ```cpp
   Equilateral * E = new Equilateral (len);
   ```
9. What is inheritance?

10. What is polymorphism?

11. What is the output of this program? Why?

```cpp
#include <iostream>
#include <sstream>
#include <vector>
#include <cmath>
using namespace std;

class A
{
    public:
        A();
        virtual string ToString () const;
        friend ostream & operator <<
            (ostream & output, const A & a);
    protected:
        int x, y;
};

class B : public A
{
    public:
        B();
        B (int m);
        string ToString () const;
    private:
        int z;
};

class C : public A
{
    public:
        C();
        C (int m);
        string ToString () const;
    private:
        float z;
};

A::A ()
{
    x = -1; y = 1;
}

string A::ToString () const
{
stringstream ss;
ss << "A : x = " << x << " and y = " << y;
return ss.str();
}

ostream & operator << (ostream & output, const A & a)
{
    output << a.ToString();
    return output;
}

B::B()
{
    x = -1; y = 1; z = 0;
}

B::B(int m)
{
    x = m; y = m*m; z = x+y;
}

string B::ToString() const
{
    stringstream ss;
    ss << "B : x = " << x << ", y = " << y << ", and z = " << z;
    return ss.str();
}

C::C()
{
    x = -1; y = 1; z = 0;
}

C::C(int m)
{
    x = m; y = m*m; z = sqrt(x+y);
}

string C::ToString() const
{
    stringstream ss;
    ss << "C : x = " << x << ", y = " << y << ", and z = " << z;
    return ss.str();
}

int main()
{
    vector<A> V1;
    vector<A *> V2;
    for (int i = -2; i < 3; i++)
    {
        B b(i);
    }
V1.push_back(b);
C c(i);
V1.push_back(c);
A * bp = new B(i);
V2.push_back(bp);
A * cp = new C(i);
V2.push_back(cp);
}
for (int i = 0; i < V1.size(); i++)
    cout << V1[i] << endl;
for (int i = 0; i < V2.size(); i++)
    cout << V2[i] << endl;
for (int i = 0; i < V2.size(); i++)
    cout << *V2[i] << endl;
return 0;