**Classes Lab**

**CSC240**

1. A(n) **CONSTRUCTOR** is used in C++ to guarantee the initialization of a class instance.
2. A constructor has the **SAME** name as the class itself.
3. Member functions are sometimes called **METHODS** in other object- oriented languages.
4. A(n) **DESTRUCTOR** is a member function that is automatically called to destroy an object.
5. To access a particular member function, the code must list the object name and the name of the function separated from each other by a **DOT-OPERATOR**.
6. A **DEFAULT** constructor has no parameters.
7. A **TILDE** precedes the destructor name in the declaration.
8. A(n) **INLINE** member function has its implementation given in the class declaration.
9. In an array of objects, if the default constructor is invoked, then it is applied to EVERY object in the array.
10. A constructor is a member function that is **ALWAYS** invoked whenever a class instance is created.

**Circles - Exercise 1:**

**Have sphere defined with initial values of 8 for the radius and (9, 10) for the center. How does this affect existing functions and code in the main function?**

It makes the line in the main method, where the center is set, unnecessary.

**Circles – Exercise 4: Exercise 4: Add a destructor to the code. It should print the message This concludes the Circles class for each object that is destroyed. How many times is this printed? Why?**

It is printed 4 times because there are 4 objects that have memory that needs to be freed up.

**Exercise 1: Why does the member function printList have a const after its name but getList does not?**

Because by declaring it a const, both const and non-const objects can call it. It is also good practice to make a function const so that accidental changes to the object’s data are avoided.