Programming Fundamentals II Sec. 600

## Assignment #1

## Due date: 6/13/22 at 11:59 pm

1. (40 points) Briefly define (one to two sentences) each of the following ten terms.

1. Classes

*A Class is a template and foundation for creating and defining objects*.

1. Objects

*An object is an instance of a class. It represents a real-world entity and has a unique identity, state and behavior.*

1. static

*Used to allow instances to share data. Can be used as static variables or static methods. Static method cannot access (i.e. instance date fields and methods) instance members of the class.*

1. this

*A keyword. Refers to a calling object and can also be used inside a constructor to invoke another constructor of the same class.*

1. Class abstraction

*Separates the class implementation from how the class is used.*

1. Class encapsulation

*Where the details of implementation are encapsulated and hidden from the user*

1. Association

*Association is the binary relationship describing activity between two classes*

1. Aggregation

*Aggregation represents an ownership between two objects*

1. Composition

*Composition represents dependence of one object on another to exist*

1. Wrapper class

*A wrapper class allows the creator to process primitive data type values as objects*

2. (15 points) Explain the purpose of a UML class diagram in the design process of an object-oriented program. Include what is specified and what is not specified in a class diagram.

(UML) Unified Modelling Language

*The purpose of the UML helps the creator see the specific fields and methods in a class. The class diagram provides the specific return type and parameters of the methods within the program. This is important in the design process because it lets the creator know the purpose of the program and imagine how the user might best understand and identify what they see.*

3. (15 points) What are the benefits of class encapsulation? Provide an example of class encapsulation.

Class encapsulation allows the creator to modify the public classes while leaving the private class the same. It also allows for the ability to call on the methods from the class and change the variables in any other class.

**Example:**

public class Faculty {  
 private String name;  
 private String title;  
  
 public Faculty() {  
 name = "John Doe";  
 title = "Professor";  
 }  
  
 public Faculty(String name, String title) {  
 this.name = name;  
 this.title = title;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
}

public class Course {  
  
 private Faculty faculty;  
 private String title;  
 private int section;  
  
 public Course() {  
 faculty = new Faculty();  
 title = "N/A";  
 section = 000;  
 }  
  
 public Course(Faculty faculty, String title, int section){  
 this.faculty = new Faculty(faculty.getName(), faculty.getTitle());  
 this.title = title;  
 this. section = section;  
 }  
  
 public Faculty getFaculty() {  
 return new Faculty(faculty.getName(), faculty.getTitle());  
 }  
  
 public String getTitle(){  
 return title;  
 }  
  
 public int getSection() {  
 return section;  
 }  
  
 public void setFaculty(Faculty faculty) {  
 this.faculty = new Faculty(faculty.getName(), faculty.getTitle());  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 public void setSection(int section) {  
 this.section = section;  
 }  
  
}

4. (15 points) What are the two major security issues with class relationships (association, aggregation, and composition), and how are both of these issues avoided?

*Association, aggregation and composition allows for direct connections when assigning references to fields.*

*Solution is to assign references to fields using deep copies*

*Another major security is when returning references provide direct access to a private class*

*Solution is to return references to fields using deep copies*

5. (15 points) Write a short program that will perform the following math operation using the BigDecimal class and its methods. The result should also be printed to the console.

