Programming Fundamentals II Sec. 600

## Assignment #2

## Due date: 6/20/22 at 11:59 pm

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

1. Inheritance

*Allows for reuse along common features*

1. Subclass (child class)

*Inherit accessible fields and methods of a superclass and may also add new fields and methods*

1. Superclass (parent class)

*General class that the subclass inherits from. Together with the subclass forms an, “is-a” relationship.*

1. Method overriding (@Override)

*Must be in different classes related by inheritance and must have same method signature.*

1. Polymorphism

*Sub class objects can be used in place of super class objects*

1. Dynamic binding

*A method can be implemented in several classes along the inheritance chain. Method is dependent on Actual type. Any information that can only be determined at run time.*

1. Declared type

*Refers to type of reference variable in Dynamic binding*

1. Actual type

*Refers to type of object being referenced in Dynamic binding*

1. ArrayList (aka a generic)

*An object that can be used to store a list of objects and cannot use primitive data types but can use wrapper classes*

1. Protected

Keyword used to access fields and methods in superclasses

2. (15 points) What are the two benefits of inheritance?

Inheritance (allowing us to reuse code through extend classes and super classes)

Inheritance allows for multiple instances of specialized classes that all inherit the fields and methods of the general class/ super class. This saves time that would be used otherwise on rewriting the same fields and methods each time in each class again and again.

The ability to define classes (i.e, subclasses) from a existing class (i.e, super class) also allows for a more comprehensive software to read and maintain.

3. (15 points) Assume Rectangle is a class that inherits from the Polygon class. Rectangle has a getArea method, but Polygon does not. Given the following reference variable declaration and object assignment:

|  |
| --- |
| Polygon poly = new Rectangle(); |

What would happen given the following line of code? Briefly explain your answer.

double area = poly.getArea();

*In this example there is a new object created, “new Rectangle”, with the reference variable, “Polygon poly”. This would allow access to the Rectangle class and its fields and methods. The “double area = poly.getArea would access the Rectangle class and print the area of the Rectangle using the parameters in the Rectangle class.*

4. (15 points) Write a short program that **adds** five numbers of your choosing to an ArrayList of Double elements, **sorts** the ArrayList, and then **prints** its contents to the console using a foreach loop.

import java.util.ArrayList;  
import java.util.Collections;  
  
public class NumberArrayList {  
 public static void main(String[] args) {  
 ArrayList<Double> numbers = new ArrayList<>();  
  
  
 numbers.add(Double.*valueOf*("07"));  
 numbers.add(Double.*valueOf*("15"));  
 numbers.add(Double.*valueOf*("17"));  
 numbers.add(Double.*valueOf*("26"));  
 numbers.add(Double.*valueOf*("21"));  
  
 System.*out*.println(numbers);  
  
 Collections.*sort*(numbers);  
  
 for (Double number : numbers) {  
 System.*out*.println(number);  
 }  
 }  
 }

5. (15 points) State the classes that can access members of another class given each of the access modifiers: public, private, and protected. Clearly distinguish between classes within the same package and classes in different packages.

*Within the same package, the access modifiers, “public”, “protected” and default(package, can be accessed by subclasses and any other classes. With the modifier, “private”, no subclasses can access it.*

*With different packages, “public” is still accessible by subclasses and any other classes. The “protected” modifier allows access to subclasses of the superclass in the same package and other packages. Other classes cannot access it. “Private” and default(package) modifiers do not allow access to anything outside the package.*