CSCI-UA.0101-002: Assignment 6 – OOP, Inheritance, Exception Handling

Due Wednesday, November 15th at 5 p.m.

Instructions:

- The project directory folder for this assignment is called A6_project_directory_NYUnetID. Rename NYUnetID with your own NYU NetID. For example, my NetID is gp2442, so I would rename my project folder "A6_project_directory_gp2442".
- The project directory contains a project directory containing four subdirectories, namely bin, lib, test and src. The source files are in src/edu/nyu/cs/NetID. Make sure to rename the subdirectory /NetID to your actual NYU NetID.
- Complete the code according to the instructions in this document as well as those written as comments within the .java source files (if any).
- **Important:** In addition to completing all problems, you are also expected to compile and run your source code using the Command line from the project directory. Refer to **Lectures 3 and 4** for how to do this. You will lose **five points** if we cannot compile and run your compiled code from the project directory.
- Submit a zip file named "A6_complete_NYUnetID" containing your project folder called "A6_project_directory_NYUnetID". Again, NYUnetID should be replaced with your NYU NetID.

1 The Triangle class.

a) Design a class named Triangle that extends the GeometricObject.java class file provided for you in the subdirectory src/edu/nyu/cs/NetID. The class Triangle contains:

- Three double data fields named **side1**, **side2**, and **side3** with default double values **1.0** to denote three sides of a triangle.
- A no-arg constructor that creates a default triangle.
- A constructor that creates a triangle with the specified **side1**, **side2**, and **side3**.
- The getter methods for all three data fields.
- A method named **getArea()** that returns the area of this triangle.
- A method named **getPerimeter()** that returns the perimeter of this triangle.
- A method named **toString()** that returns a string description for the triangle.

b) Write a test program in the source file TestTriangle.java that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should:

- Create a **Triangle** object with these sides and set the **color** and **filled** properties (inherited from the GeometricObject class) using the input.
- The program must display the area, perimeter, color, and true or false to indicate whether it is filled or not.
- Please make sure the output is clear and easy to read.

2 Writing a class file using a UML class diagram.

A Unified Modeling Language (UML) class diagram is a type of diagram that describes the structure of an object-oriented class. The UML class diagram takes the form

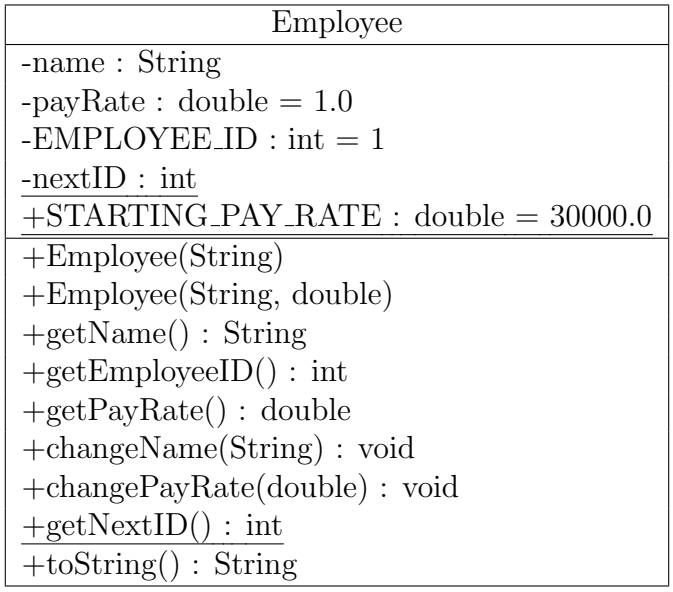
ClassName
vis attribute : type
vis operation(arg list) : return type

where the fields stand for: vis = visibility (+ for public, - for private), attribute = data member and operation = method or constructor.

Note the following:

- The arg list is a list of parameter types (e.g., int, double, String); parameter names are **not** included in the UML class diagram.
- Methods that don't return a value (i.e., void methods) should give a return type of void
- Static methods and fields are indicated by underlining.
- Constant (i.e., final) fields should be in ALL_CAPS. The final keyword is declared before the return type, and after the static keyword (if used at all).

With this information, complete the Employee source file, which is provided in the src/edu/nyu/cs/NetID subdirectory, as specified by the UML Diagram below. The toString() method should return the string "Employee's name: ?", where ? is the name of the employee.


Employee
-name : String
-payRate : double = 1.0
-EMPLOYEE_ID : int = 1
-nextID : int
+STARTING_PAY_RATE : double = 30000.0
+Employee(String)
+Employee(String, double)
+getName() : String
+getEmployeeID() : int
+getPayRate() : double
+changeName(String) : void
+changePayRate(double) : void
+getNextID() : int
+toString() : String

3. Exception handling.

a) Write a program that meets the following requirements:

- The program should be saved in a source file called "**Ex2a**", without the quotes.
- The program creates an array with 100 randomly chosen integers.
- The program prompts the user to enter the index of an array, then displays the corresponding element value. If the specified index is out of bounds, display the message "Out of Bounds", without the quotes.

b) In Question 1, you defined the **Triangle** class with three sides. In a real triangle, the sum of any two sides is greater than the other side. Here you'll modify the **Triangle** class to adhere to this rule. Create the **IllegalTriangle-Exception** class, and modify the constructor of the Triangle class to throw an IllegalTriangleException object if a triangle is created with sides that violate the rule. You are free to decide how to handle the exception, but in addition to how to handle the exception, you **must** print out the stack trace of the Exception by calling the .printStackTrace() method.