COMP1161 LAB1- The AssignPlan

PREAMBLE

This lab aims to reinforce the concepts of writing a simple Java class. The specific skills to be covered are:

1. Declaring a class
2. Declaring attributes
3. Writing constructors
4. Writing accessors and mutators
5. Evaluating expressions in Java

Your programs will be implemented on HackerRank at the following link:
http://www.hackerrank.com/comp1161-lab1-23assign

Imagine a guideline is developed that advises students on how to manage assignments. A student that uses guideline will create an **AssignPlan** each day, that will manage summarized data about their assignments. While a student is supposed to plan how to complete all assignments, if there are any urgent assignments, they will be handled immediately. For this lab, we will make the simplifying assumption that all assignments can be done within a day (ie. No assignment will be carried forward to another day). Also, note that while a real implementation may associate each **Assignment** with an **AssignPlan**, for this lab we are not concerned with coding the associations.

MARKING CRITERIA (PLEASE REVIEW BEFORE STARTING)

Criterion	Mark(s)
Class name is specified	1
Attributes are properly declared	1
Constructors are written as specified	1
Accessors and Mutators are written as specified	1
Tests demonstrate that the code works	1
A PENALTY OF 1 MARK WILL BE APPLIED FOR ANY OF THE FOLLOWING: • Failure to declare attributes as private • Failure to spell method names as specified • Writing public methods other than those specified.	

LAB EXERCISES

You are required to complete a partially written Java class that is expected to represent an **AssignPlan**. Your expected tasks are :

1. Create accessor methods for the private variables **numAssignments**, **numComplete**, **totalPoints**, **hoursAvailable** and **highestPriorityItem**.
2. Create an alternative constructor that accepts and sets **numAssignments**, **numComplete**, **totalPoints**, and **highestPriorityItem**.
3. Implement a mutator function to set **highestPriorityItem**.
4. Implement a class to represent an **Assignment**.
    a. The **Assignment** class will privately store information on the effort (hours required to complete)*(int)*, the number of resources to be referenced*(int)*, the estimated difficulty*(int)*, and the expected score*(double). (Decide on your own variable names).*
    b. Create a constructor for a **Assignment**, that accepts the effort, resources and difficulty.
    c. Create a public method that evaluates and returns the expected score, using the formula below, which was supplied by a "well paid consultant".

$$\boldsymbol{estScore} = \mathbf{0.1} * (\boldsymbol{effort} * \boldsymbol{difficulty} + \frac{(\boldsymbol{resources} * \boldsymbol{effort})^2}{\boldsymbol{\pi}\sqrt{\boldsymbol{difficulty}}})$$

    (Note: Use the **PI** attribute of the **Math** class)
    Also, report the score to 1 decimal point (important for passing test cases 4 and 5)

    d. Create accessors for all private variables.

5. Complete the **handleUrgentAssignment** method of the **AssignPlan** object so that it
    a. Sets the highestPriorityItem to the name of the new assignment being handled.
    b. Creates an **Assignment** with knowledge of the difficulty, resources and effort
    c. Increment the number of assignments in the **AssignPlan** by 1
    d. Increments the **numComplete** by 1
    e. Update the **hoursAvailable** by subtracting the **effortHours**
    f. Update the total points by adding the points calculated on the assignment.