**Exercise 1**

The driver class for this exercise will be provided to you. You have to define the classes and their relationships from the below given UML diagram. Write comments for all constructors, methods, and generate Java docs.

**Note:** The name of the project will be your “*Lastname*\_Lab06AbstractAndInterfaces”. For example “Smith\_ Lab06AbstractAndInterfaces”.

All classes must be placed in a package named “**lab06abstractandinterfaces**”

1. Define the interface **Employee** specified in the UML diagram below.

|  |
| --- |
| *<<Employee>>* |
|  |
| + getWeeklySalary():double |

2. Define the interface **Employer** specified in the UML diagram below.

|  |
| --- |
| *<<Employer>>* |
|  |
| +getNumOfEmployees():int |
| +setNumOfEmployees(number : int):void |

3. Define the abstract class **AbstractStudent** specified in the UML diagram below.

|  |
| --- |
| *AbstractStudent* |
| -nameOfStudent:String |
| -annualTuition: double |
| +AbstractStudent(nameOfStudent:String, annualTuition:double)  +getAnnualTuition( ):double  +getTuition(): double // an abstract method  +toString():String |

The constructor **AbstractStudent** initializes the attribute nameOfStudent and annualTution.

The method **getAnnualTuition** returns annual tuition.

The method **getTuition** is an abstract method.

The method **toString** returns a string that has the name of the student followed by the tuition he pays every year. The name prints in a field of length 20 and the tuition is a floating-point value, so print it with two decimal places in the field of length 10.

4. Define the class Student that extends **AbstractStudent** and implements the interfaces **Employee** and **Employer** specified in the UML diagram below.

|  |
| --- |
| Student |
| -nameOfInstitution:String |
| -hoursWorked:int |
| -hourlyRate:double |
| -numOfEmployees:int |
| -years:int |
| +Student(nameOfStudent:String, annualTuition:double, years:int, nameOfInstitution: String, hoursWorked:int, hourlyRate:double,numOfEmployees:int)  +toString():String |

Note: hoursWorked is for single day.

The constructor **Student** initializes the attributes declared in the class Student. For the attributes declared in its superclass, your program should invoke the superclass constructor to initialize them.

The method **toString** returns the values of all attributes in the same order they are declared. Specifically, this **toString** method should call the **toString** defined in the class **AbstractStudent** and then print the values of the attributes which are additionally added to the class **Student**. The values are separated by one space. For strings, print each of them in the field of length 20; for integers, print each of them in the field of length 4; for floating-point values, print each of them in the field of length 10 with two decimal places.

After you create the class Student, you will see errors. You can fix them by defining the methods

+getTuition(): double

+getWeeklySalary():double

+getNumOfEmployees():int

+setNumOfEmployees(number : int)

**getTuition** returns the multiplication of the number of years a student needs to study for his/her degree by the annualTuition.

**getWeeklySalary** returns the multiplication of the hourlyRate by the number of hours the student worked per week.

**getNumOfEmployees** returns the value of the attribute numOfEmployees.

**setNumOfEmployees** sets the value of the attribute numOfEmployees.

5. Copy the class **StudentDriver** into your project first and then uncomment the code present in the main method. You will see many errors. Explain each error in the comment after the statement where the error occurs with your own words. Don’t copy the error messages on Netbeans directly.

6. Comment out all the statements with errors without fixing them. Run the project, you should be able to see the output as displayed below.

Sample Output:

Print Student Type Reference Variable

Calton 2250.00 3 NWMSU 10 3.65 10

Print AbstractStudent Type Reference Variable

Alex 2300.00 3 UHCL 14 7.65 5

Print Employee Type Reference Variable

Kara 3600.00 2 UCM 12 8.85 20

Print Employer Type Reference Variable

Adam 3250.00 4 UMKC 9 4.32 50

Using Student Type Reference Variable

Total Tuition Fee: $6750.00

Salary for one week: $255.50

Number of Employees: 10

Updated number of Employees: 22

Using AbstractStudent Type Reference Variable

Total Tuition Fee: $6900.00

Using Employee Type Reference Variable

Salary for one week: $743.40

Using Employer Type Reference Variable

Number of Employees: 50

Updated number of Employees: 60