Homework 08: Inheritance and Exceptions

Release date: 3/4
Due date: 3/11 11:59 PM

Goals:

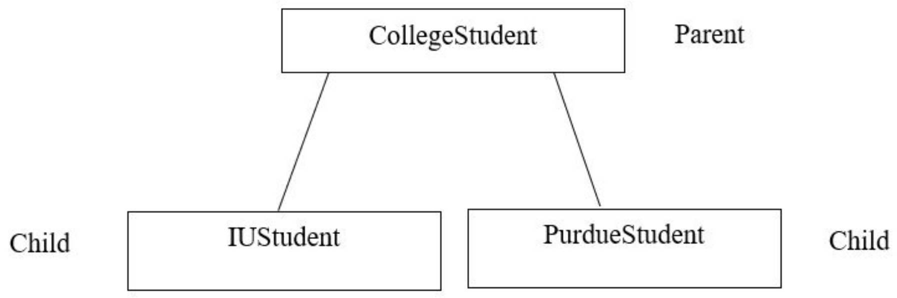
- Become familiar with subclasses and superclasses
- Learn how to override methods
- Learn how to use inherited constructors and methods

Introduction:

The main advantage of inheritance is to reduce the amount of work and code needed by reusing existing code in different classes. As the lab for inheritance and exceptions has stated, this is achieved by saying one class **`extends`** another class, on which the former class inherits all public and protected variables and methods of the latter class. Again, the subclass, the class that inherits the methods and variables from the superclass, can override methods using the same method name to provide its own implementation of that method.

Exceptions are events that disrupt the flow of a program and can be thrown unintentionally or intentionally. Unintentional exceptions are usually runtime exceptions thrown by the JVM and are unchecked at compile time. Some examples include ArithmeticException and ArrayIndexOutOfBoundsException. Intentional exceptions are exceptions that the user can throw to indicate an error. These can be custom exceptions or existing runtime exceptions with custom messages.

In this homework, you will create 3 classes: **CollegeStudent**, **IUStudent**, and **PurdueStudent**. **CollegeStudent** will be the parent class and **IUStudent** and **PurdueStudent** will extend **CollegeStudent**. At certain times, an IllegalArgumentException with a custom message with be thrown. The details about the instance variables and methods of each class are detailed below.



Task 0: Residency Enum

Because we will only have three different residency statuses (in-state, out of state, and international), it's reasonable that we create an enum to represent these. Create an enum named **Residency** with the values ***IN_STATE***, ***OUT_OF_STATE***, and ***INTERNATIONAL***, listed in that order.

Task 1: CollegeStudent

Fields

All fields are private and non static.

Name	Type
`dormCost`	`double`
`tuition`	`double`
`residency`	`Residency`
`livesOffCampus`	`boolean`

Constructors

Name	Return Type	Parameters
`CollegeStudent()`	`CollegeStudent`	`- String residency`

`CollegeStudent()`	`CollegeStudent`	`- String residency`
		`- boolean livesOffCampus`

Methods

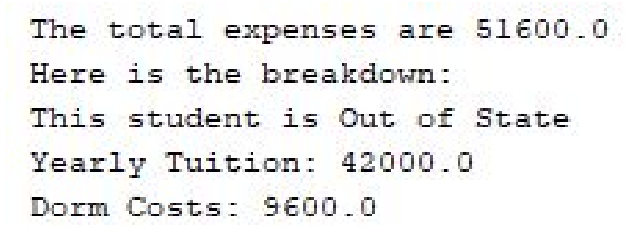
Name	Return Type	Parameters
`setTuition()`	`void`	`- double tuition`
`setDormCost()`	`void`	`- double dormCost`
`calculateYearlyCost()`	`double`	`(None)`
`getTuition()`	`double`	`(None)`
`getDormCost()`	`double`	`(None)`

`getResidency()`	**`Residency`**	**`(None)`**
`isLivingOffCampus()`	**`boolean`**	**`(None)`**
`toString()`	**`String`**	**`(None)`**

Information about Fields/Constructors/Methods

Name	Description
`tuition`	Corresponds to the cost of attending college for one year. By default, international students' tuition is $44,500, out of state students' tuition is $42,000, and in state students' tuition is $23,000.
`residency`	Note that the field is an enum, but the parameter residency in the constructors is a String. You should only accept **`Strings`** that are spelled like the corresponding enum values (case insensitive) with spaces instead of the underscores. Any other Strings should result in an **`IllegalArgumentException`** with the message "Student residency must be one of the three specified statuses."
`dormCost`	The default cost of living on campus is $800 **per month**, and off campus is $500 **per month**.
`livesOffCampus`	If not specified in the constructor parameters, this will be **`false`**
`CollegeStudent()`	Responsible for initializing each of the student's fields.
`calculateYearlyCost()`	The total cost of tuition added with the dorm cost for one year.
`getDormCost()`	Returns the monthly cost of living.
`toString()`	Returns information about the class as a string. (See output below). We will not grade the formatting of the String, but rather the numbers given. Note that dorm costs are given as the yearly cost.

Example toString() Output



```
The total expenses are 51600.0
Here is the breakdown:
This student is Out of State
Yearly Tuition: 42000.0
Dorm Costs: 9600.0
```

Task 2: IUStudent

Fields

All fields are private and non static.

Name	Type
`bookFees`	`double`
`transportationCost`	`double`
`financialAid`	`double`

Constructor

Name	Return Type	Parameters

`IUStudent()`	`IUStudent`	`- String residency`
		`- boolean livesOffcampus`
		`- double GPA`

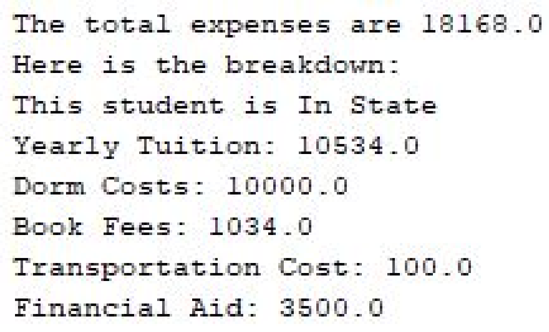
Methods

Name	Return Type	Parameters
`setTuition()`	`void`	`- double tuition`
`setDormCost()`	`void`	`- double dormCost`
`setTransportationCost()`	`void`	`- double transportationCost`
`setBookFees()`	`void`	`- double bookFees`
`setFinancialAid()`	`void`	`- double financialAid`
`getBookFees()`	`double`	`(None)`
`getFinancialAid()`	`double`	`(None)`
`getTransportationCost()`	`double`	`(None)`
`calculateYearlyCost()`	`double`	`(None)`
`toString()`	`String`	`(None)`

Information about Fields/Constructors/Methods

Name	**Description**
`bookFees`	Books will cost $1,034 by default.
`transportationCost`	If a student lives off campus, they will pay $500 in transportation costs per year. Else, they will pay $100 in transportation costs per year.
`financialAid`	Set **`financialAid`** based on **`GPA`**. $4,500 is given for a minimum GPA of 3.75, $3,500 for a minimum of 3.50, $2,750 for a minimum of 3.00, and $2,500 for a minimum of 2.50. These values are not cumulative (a student with a 3.10 will not receive $2,750 AND $2,500).
`tuition`	Out of state students' tuition is $34,846 and in state students' tuition is $10,534.
`dormCosts`	The default cost of living on campus is $800 **per month**, and off campus is $400 **per month**.
`IUStudent()`	Sets the fields for both this class and the parent class. If GPA is a negative number, an **`IllegalArgumentException`** is thrown with the message: "GPA needs to be above or equal to 0". If a student's residency is international or any other value,an **`IllegalArgumentException`** is thrown with the message: "Student must be in state or out of state".
`calculateYearlyCost()`	Sums tuition, book fees, transportation costs, dorm costs, and deducts financial aid.
`toString()`	Returns information about the class as a string. (See output below)

Example Output of toString()



```
The total expenses are 18168.0
Here is the breakdown:
This student is In State
Yearly Tuition: 10534.0
Dorm Costs: 10000.0
Book Fees: 1034.0
Transportation Cost: 100.0
Financial Aid: 3500.0
```

Task 3: PurdueStudent

Fields

All fields are private and non static.

Name	Type
`bookFees`	`double`
`financialAid`	`double`
`major`	`String`

Constructor

Name	Return Type	Parameters

`PurdueStudent()`	`PurdueStudent`	`- String residency`
		`- String major`
		`- boolean livesOffCampus`
		`- double GPA`

Methods

Name	Return Type	Parameters
`setTuition()`	`void`	`- double tuition`
`setBookFees()`	`void`	`- double bookFees`

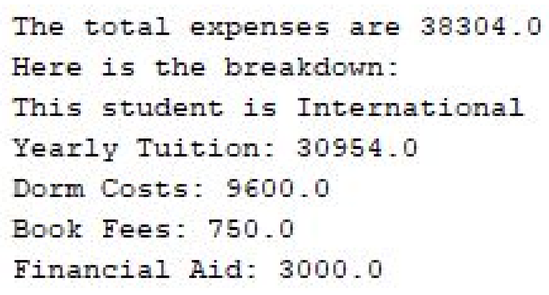
`setFinancialAid()`	**`void`**	**`- double financialAid`**
`getBookFees()`	**`double`**	**`(None)`**
`getFinancialAid()`	**`double`**	**`(None)`**
`getMajor()`	**`String`**	**`(None)`**
`calculateYearlyCost()`	**`double`**	**`(None)`**
`toString()`	**`String`**	**`(None)`**

Information about Fields/Constructors/Methods

Name	**Description**
`tuition`	International students' tuition is $30,954, out of state students' tuition is $28,794 and in state students' tuition is $9,992.
`bookFees`	Set bookFees based on major (case insensitive). "Computer Science" students pay $200, "Engineering" students pay $500, "Liberal Arts" students pay $750, and any other students pay $100.
`financialAid`	Sets **`financialAid`** based on **`GPA`**. $5,000 is given at a minimum GPA of 3.75, $4,500 for a minimum of 3.50, $3,000 for a minimum of 3.00, and $2,500 for a minimum of 2.50. These values are not cumulative.
`dormCosts`	**`dormCosts`** should behave similarly to **CollegeStudent**'s dorm costs.
`PurdueStudent()`	Sets the fields for both this class and the parent class. If GPA is a less than 2, an **`IllegalArgumentException`** is thrown with the message: "GPA needs to be above or equal to a 2.00".
`calculateYearlyCost()`	Sums book fees, tuition, dorm costs and deducts the amount of given by financial aid.

`toString()`	Returns information about the class as a string. (See output below). We will be looking for the values in the String rather than the format of the String.

Example Output of toString()



```
The total expenses are 38304.0
Here is the breakdown:
This student is International
Yearly Tuition: 30954.0
Dorm Costs: 9600.0
Book Fees: 750.0
Financial Aid: 3000.0
```

Note: It is a good idea to create a main method and create objects of **CollegeStudent**, **IUStudent**, and **PurdueStudent** to test your code.

Skeleton Code

There will be no starter code for this homework assignment.

Submission

Submit the following files to BlackBoard:

- **CollegeStudent.java**
- **IUStudent.java**
- **PurdueStudent.java**
- **Residency.java**

Keep in mind that only your latest submission will be considered.

Rubric

- **Static and non-private variables**: If you use a static or non-private variable, you will receive 0 points.
- **CollegeStudent** class - 40 points total
  - -Correct constructors that initialize the instance variables - 15 points
  - -Throws an IllegalArgumentException when necessary - 5 points
  - -Correct calculation of total expenses - 15 points
  - -Correct toString() - 5 points
- **IUStudent** class - 30 points total
  - -Correctly inherits from CollegeStudent - 5 points
  - -Correct constructors that initialize the instance variables - 5 points
  - -Correct calculation of total expenses - 5 points
  - -Correct setter methods - 5 points
  - -Throws an IllegalArgumentException when necessary - 5 points
  - -Correct toString() - 5 points

- **PurdueStudent** class - 30 points total
  - -Correctly inherits from CollegeStudent - 5 points
  - -Correct constructors that initialize the instance variables - 5 points
  - -Correct calculation of total expenses - 5 points
  - -Correct setter methods - 5 points
  - -Throws an IllegalArgumentException when necessary - 5 points
  - -Correct toString() - 5 points