Assignment 1: Program StudentMan

1. Requirement

Program StudentMan of a university captures and processes data about an entity named `Student` and two specific types of `Student` named `UndergradStudent` (undergraduate student) and `PostgradStudent` (postgraduate student). Figure 1 shows the concept class diagram of StudentMan. Table 1 lists the attribute design details of the three classes.

To ease listing of customer objects, program StudentMan requires that `Student` realises an interface named `Comparable`. This interface is provided by Java. The implementation of the method `Comparable.compareTo` must compare two `Students` by name. Please note that your program will be marked automatically by a program, which expects you to **use annotation** and strictly follow the essential the design rules. An error in one part may affect other parts.

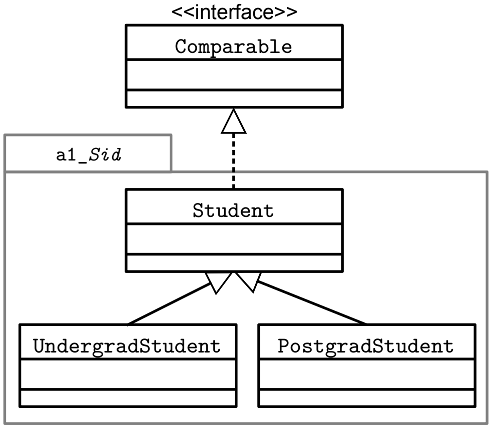

Figure 1: Class diagram.

Table 1: The attribute design details of program classes.

Attribute(s)	formal type	mutable	optional	min	max	length
`id`	`Integer`	F	F	for `Student`		
				1	10^9	-
				`UndergradStudent`		
				10^5	10^8	-
				`PostgradStudent`		
				$10^8 + 1$	10^9	-
`name`	`String`	T	F	-	-	50
`phoneNumber`	`String`	T	F	-	-	10
`address`	`String`	T	F	-	-	100
***Specific to* `PostgradStudent`**						
`gpa`	`Float`	T	F	0.0	4.0	-

2. Tasks

1. Create a package named `a1_`*`Sid`* as shown in Figure 1, where *`Sid`* is your student id. For example, if your student id is 123456789 then the package name is `a1_123456789`. You will need to use this package to hold all the Java classes that you create for the program.

   IMPORTANT: Package `utils` *must not* be created as a sub-package of package `a1_`*`Sid`*.

Failure to name the package as described will result in an invalid program.

2. Specify and implement class `Student` with all the ***essential*** attribute(s) and operations.

   Note: this class must use exceptions where suitable to increase code robustness.

3. Specify and implement class `UndergradStudent` with all the ***essential*** attribute(s) and operations.

   Note: this class must use exceptions where suitable to increase code robustness.

4. Specify and implement class `PostgradStudent` with all the ***essential*** attribute(s) and operations.

   Note: this class must use exceptions where suitable to increase code robustness.

IMPORTANT: The essential mutator and observer operations of the same attribute affect one another. An error in one operation affects the validity of the other operation.

3. Submission

Create a zip-compressed file containing **just the folder of the package** specified in Task Error: Reference source not found. You must name the file as follows: **a1_*Sid*.zip**, where ***Sid*** is your student id.

Submit your file to the designated submission box for this assignment.

IMPORTANT: Failure to name the file as described above will result in an invalid program. In particular, ONLY the **ZIP** format is accepted. Other formats are NOT accepted.