Lab 3

Deadline: 20 September 2020, 10 PM

Objectives
- Familiar with Inheritance, Polymorphism and interface.

Task 1:

The student record application is described as follows.

Student class (abstract class)
- Name (first and last name)
- DOB
- Address
- Mobile number
- Student ID

Undergraduate Student class
- Name (first and last name)
- DOB
- Address
- Mobile number
- Student ID
- Current course
- Expected completed session (session, year)

Post-Graduate Student class
- Name (first and last name)
- DOB (date)
- Address
- Mobile number
- Student ID
- Current course
- Expected completed session (session, year)
- Previous degree
- Date of Completion of the previous degree (date)

Address class
- Street number
- Street name
- City
- State
- Postcode
- Country

Student System class
- students (ArrayList<Student>)

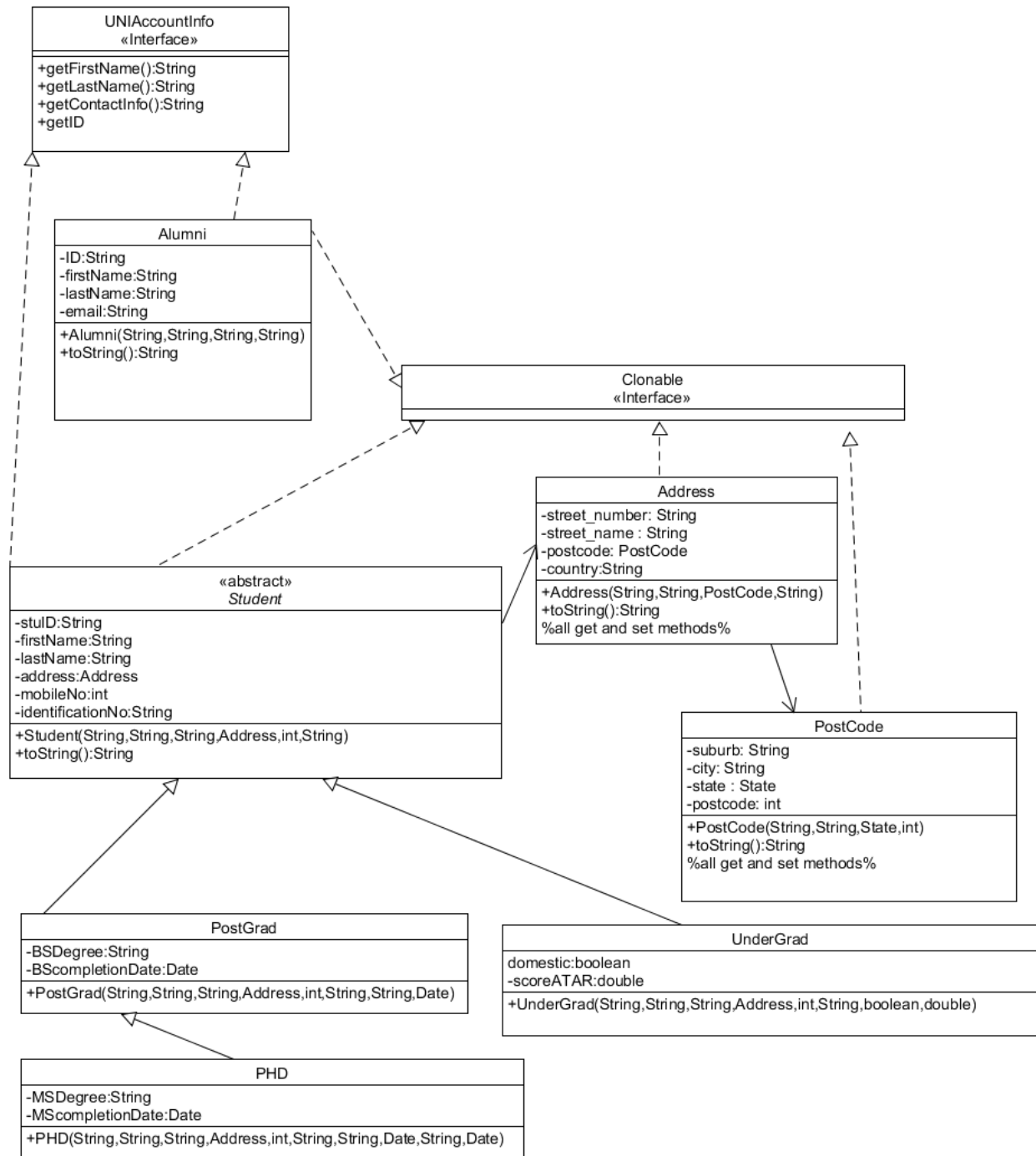Test Code class
- Test the program (main())

Draw UML class diagrams and write java program with the below instructions.

1. From the above information, draw the UML class diagram of the classes and the relationship between the classes.
2. Write java program to construct the above class. Design your own methods for each class using the concept of OOP design principle.
3. Design a StudentSystem class that keep the record of all students (i.e., an Arraylist of Student objects). In the StudentSystem class, design a student search method that takes an int (Student id) as the input and print all information of the student (overwrite toString() method for the printing).
4. Write test code in the main method in the TestCode Class with the following steps:
   a. create an object of StudentSystem;
   b. create one undergraduate student (Student ID:100000) and one postgraduate student (Student ID:200000) objects and add the two student objects to the student arraylist;
   c. use the student search method to search and print (using toString() and polymorphism) the information of the two students;
   d. use a for loop to go through the student arraylist and update the undergraduate student's expected completion session to "Spring 2022", and update the postgraduate student's previous degree to "Bachelor of Math" (using instanceof and downcasting);
   e. use another for loop to print the updated information of the two students.

Task 2:

The University online shopping members system is briefly described in the below UML class diagram. Please complete the UML class diagram and write a java program following the below instructions.

1. Write all the code to construct the below class in the diagram. Design your own methods for each class using the concept of OOP design principle.
2. Implement all clone() methods (deep clone) for all classes that implements Clonable interface.
3. Write test code in the main method in the TestCode Class and create a PhD object and a Undergraduate object and testing the clone() methods with this two objects.

UNIAccountInfo
«Interface»

+getFirstName():String
+getLastName():String
+getContactInfo():String
+getID

Alumni

-ID:String
-firstName:String
-lastName:String
-email:String

+Alumni(String,String,String,String)
+toString():String

Clonable
«Interface»

«abstract»
Student

-stuID:String
-firstName:String
-lastName:String
-address:Address
-mobileNo:int
-identificationNo:String

+Student(String,String,String,Address,int,String)
+toString():String

Address

-street_number: String
-street_name : String
-postcode: PostCode
-country:String

+Address(String,String,PostCode,String)
+toString():String
%all get and set methods%

PostCode

-suburb: String
-city: String
-state : State
-postcode: int

+PostCode(String,String,State,int)
+toString():String
%all get and set methods%

PostGrad

-BSDegree:String
-BScompletionDate:Date

+PostGrad(String,String,String,Address,int,String,String,Date)

UnderGrad

domestic:boolean
-scoreATAR:double

+UnderGrad(String,String,String,Address,int,String,boolean,double)

PHD

-MSDegree:String
-MScompletionDate:Date

+PHD(String,String,String,Address,int,String,String,Date,String,Date)

Submission:

- Please submit your solution to Moodle. Email submission is not accepted.
- Please submit an **individual PDF document** to contains all your solutions for the two tasks.
- For both tasks, please include your UML class diagrams, java program and the snapshots which clearly shows the compilation and the execution of your program.

Note: Plagiarism will be reported to the school, and all involving students will receive the zero mark.