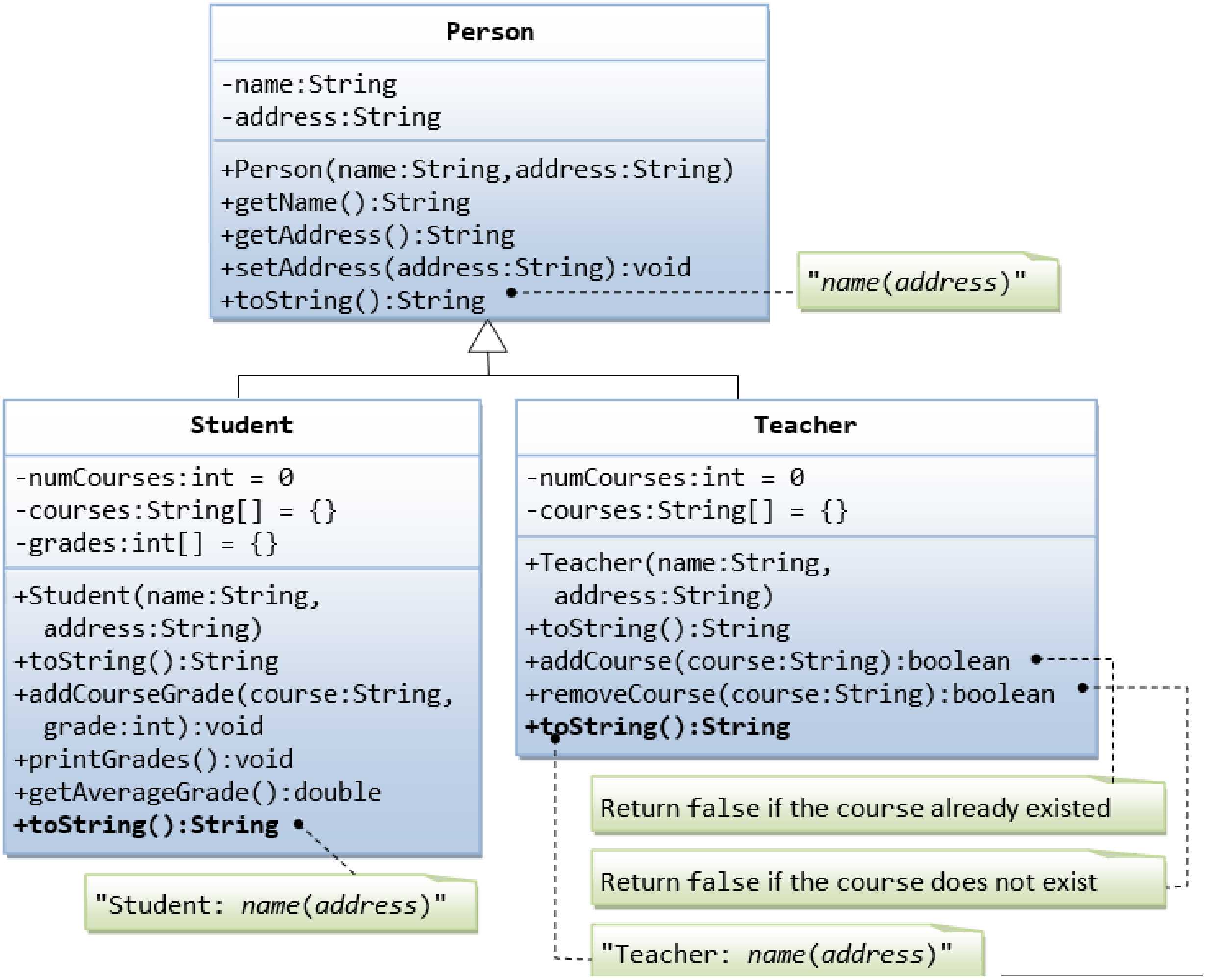
UNIVERSITY OF RUHUNA
BACHELOR OF COMPUTER SCIENCE (BCS) (GENERAL) DEGREE
LEVEL II (SEMESTER I)
Laboratory Assignment 05

CSC 2123 – Object Oriented Programming **TIME:3 Hours**

1. Consider the following super class and sub class relationship.


Person
-name:String
-address:String
+Person(name:String,address:String)
+getName():String
+getAddress():String
+setAddress(address:String):void
+toString():String
"name(address)"
Student
-numCourses:int = 0
-courses:String[] = {}
-grades:int[] = {}
+Student(name:String, address:String)
+toString():String
+addCourseGrade(course:String, grade:int):void
+printGrades():void
+getAverageGrade():double
+toString():String
"Student: name(address)"
Teacher
-numCourses:int = 0
-courses:String[] = {}
+Teacher(name:String, address:String)
+toString():String
+addCourse(course:String):boolean
+removeCourse(course:String):boolean
+toString():String
Return false if the course already existed
Return false if the course does not exist
"Teacher: name(address)"

Person class contains,

Two private variables, name(String), address (String) ,

A constructor that initialize name and the address,

Getters getName(), getAddress()

returns name and address respectively.

Setters setAddress(String adress)

Set address value to the variable.

toString():

returns the name and the address.

Student class contains,

Three private variables,

numCourses(int)- number of courses taken so far,

courses String[] –Course Codes,

grades int[]- grade for the corresponding course codes

A constructor that initialize name and the address,

addCourseGrade()-void

Add a course and its grade - No validation in this method,

numCourses variable will be incremented.

printGrades()-void

Print all courses taken and their grade.

getAverageGrade()-double

Compute the average grade

toString():

override the super class method and print student name and address.

Teacher class contains,

Two private variables,

numCourses(int)- number of courses taught currently,

courses String[] –Course Codes,

A constructor that initialize name and the address,

addCourse(String course)-boolean

Return false if the course already existed

First Check if the course already in the course list

If not, add the course,

numCourses variable will be incremented.

removeCourse(String course)-boolean

Return false if the course cannot be found in the course list.

First find out the corresponding course index, if the course is found, then remove the course and rearrange the course array.

getAverageGrade()-double

Compute the average grade

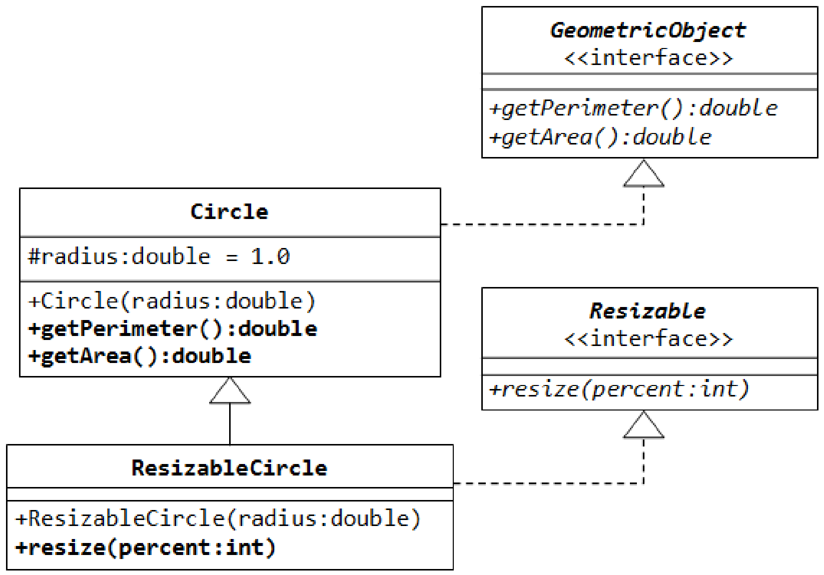
toString():

override the super class method and print teachers' name and address.

i. Implement the Person, Student and Teacher classes according to the above description.

ii. Test the classes written above using a test class called TestPerson.

2. ✓



i. Write the interface called GeometricObject, which declares two abstract methods: getPerimeter() and getArea(), as specified in the class diagram.

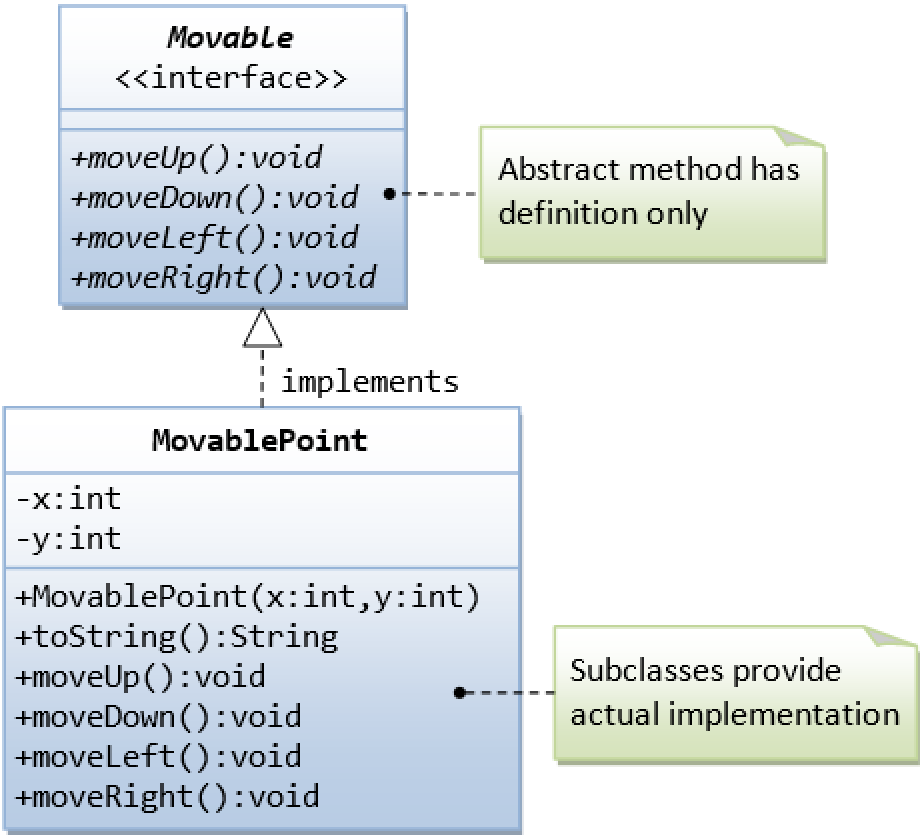
ii. Write the implementation class Circle, with a protected variable radius, which implements the interface GeometricObject.

iii. Write a test program called TestCircle to test the methods defined in Circle.

iv. The class ResizableCircle is defined as a subclass of the class Circle, which also implements an interface called Resizable, as shown in class diagram. The interface Resizable declares an abstract method resize(), which modifies the dimension (such as radius) by the given percentage. Write the interface Resizable and the class ResizableCircle.

v. Write a test program called TestResizableCircle to test the methods defined in ResizableCircle.

3. ✓



i. Write the interface called Movable, which declares four abstract methods: moveUp(), moveDown(), moveLeft() and moveRight(), as specified in the class diagram.

ii. Write the implementation class MovablePoint, with private variables x and y ((x, y) coordinates of the point), which implements the interface Movable.

iii. Write a test program called TestMovablePoint to test the methods defined in MovablePoint.

4. Create Java program templates to the following description.

System: Computer Order System

Description: A company has a business to sell a product through order. To order a book, we must become a customer of the company. As a customer, we have to submit our information such as national id, name, address, and phone number. When we become a customer, we could make an order to the company. In each order, it will be printed order id, order date, the customer who makes an order, the employee (it has id, name, and title) who handles the order, and the list of the products. Products are classified into three categories book, software, and hardware. Each product will be given code, name, price, and stock. The Book has additional information such as the author name, publisher, and year. The software will have additional information such as; operating system, company, and version. The hardware consists of type, and brand.

5. Create Java Program templates to represent the following requirement.
   System: Bank Account
   Description: A specific bank would like to develop application for the account system. In the system, there are customers who have information such as: National_Id, Name, Phone, and Address. Each customer can have many accounts. The account has the following information Account_Id, Balance. There are four types of accounts: Current_Account, Checking_Account, Saving_Account, and Join_Account. Each Current_Account will be given a ATM_CARD which has Card_Number and Pin, each Checking_Account will be given a set of check_book which has the prefix_number, and amount, each Saving_Account will be given the commission based on the balance and company profits, and each Joint_Account will have list of customers who are registered in this account.

6. ✓ Write a simple java function to accept two integer values and divide two numbers and name it as doDivide(). Then write a java program to get inputs to doDivide() function from users and display the output. You should handle any arithmetic exception that can arise when executing the function.

7. Suppose a company is maintaining a discount management system which will calculate discounts for the total bill of the purchases. If the total bill is greater than 10,000 rupees then 5% of discount for the total bill will be applied. If the total bill is greater than 20,000 rupees then 10% discount will be applied for the final bill. Discount is given up to maximum 50,000 rupees. Create a custom exception to display an error message if the total bill is less than 10,000 rupees and bill is greater than 50,000 rupees.

****