CS 501B – Introduction to JAVA Programming
Fall 2022 Semester
Due: 10/14/2022 Friday at 11:59 PM

Instructions:

1) You are not allowed to use any package/library unless stated otherwise.
2) Please comment your name and CWID in the first two lines of every .java file.
3) You are required to zip the all the .java files as FirstName_LastName_Assignment#.zip (Ex: Suhas_HS_Assignment2.zip).
4) This assignment covers topics from weeks 5 and 6.
5) Students are not allowed to collaborate with classmates and any other people outside. All work must be done individually. Any work having evidence of showing academic dishonesty violation are subjected to give zero for the assignment.

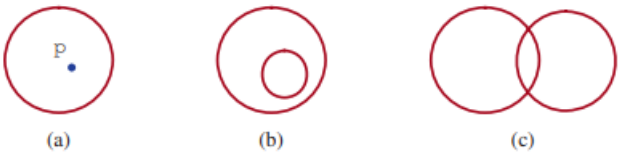
Problems:

1) (40 Points) Define the MyRectangle2D class that contains:
   a) Two double data fields named x and y that specify the centre of the rectangle with public getter (getX(), getY()) and public setter (setX(double newX), setY(double newY)) methods. (Assume the rectangle sides are parallel to x- or y-axis.).
   b) The double data fields width and height with public getter (getWidth(), getHeight()) and public setter methods (setWidth(double w), setHeight(double h)).
   c) A no-arg constructor that creates a default rectangle with (0, 0) for (x, y) and 1 for both width and height.
   d) A constructor that creates a rectangle with the specified x, y, width, and height.
   e) A public method getArea() that returns the area of the rectangle.
   f) A public method getPerimeter() that returns the perimeter of the rectangle.
   g) A public method contains(double x, double y) that returns true if the specified point (x, y) is inside this rectangle (Fig (a)).
   h) A method contains(MyRectangle2D r) that returns true if the specified rectangle is inside this rectangle (Fig (b)).
   i) A method overlaps(MyRectangle2D r) that returns true if the specified rectangle overlaps with this rectangle (Fig (c)).

(a) (b) (c)

2) (30 Points) Define the Circle2D class that contains:
   a) Two double data fields named x and y that specify the center of the circle with public getter methods (getX() and getY()).
   b) A data field radius with a public getter method (getRadius()).
   c) A no-arg constructor that creates a default circle with (0, 0) for (x, y) and 1 for radius.
   d) A constructor that creates a circle with the specified x, y, and radius.
   e) A public method getArea() that returns the area of the circle.
   f) A public method getPerimeter() that returns the perimeter of the circle.
   g) A public method contains(double x, double y) that returns true if the specified point (x, y) is inside this circle (Fig (a)).
   h) A public method contains(Circle2D circle) that returns true if the specified circle is inside this circle (Fig (b)).

i) A public method overlaps(Circle2D circle) that returns true if the specified circle overlaps with this circle (Fig (c)).



3) (10 Points) Design a class named Account that contains:
   a) A private int data field named id for the account (default 0).
   b) A private double data field named balance for the account (default 0).
   c) A private double data field named annualInterestRate that stores the current interest rate (default 9). Assume that all accounts have the same interest rate.
   d) A private Date data field named dateCreated that stores the date when the account was created. (Hint: Use Date library)
   e) A no-arg constructor that creates a default account.
   f) A constructor that creates an account with the specified id and initial balance. This method should throw an Exception if id or balance is negative.
   g) The public accessor and mutator methods for id (getId(), setId(int newId)), balance (getBalance(), setBalance(double amount)), and annualInterestRate (getInterest(), setInterest(double interest)). The setter methods should throw an Exception if the ID, amount or interest is negative.
   h) The public accessor method for dateCreated (getDate()).

4) (20 Points) Design a AccountNew class that extends Account class as follows:

   a) Add a new data field name of the String type to store the name of the customer.
   b) Add a new constructor that constructs an account with the specified name, id and balance.
   c) Add a new public method named getMonthlyInterestRate() that returns the monthly interest rate.
   d) Add a new public method named getMonthlyInterest() that returns the monthly interest.
   e) Add a new public method named withdraw(double amount) that withdraws a specified amount from the account. If the withdrawal amount is greater than the account balance, the method should throw an Exception.
   f) Add a new public method named deposit(double amount) that deposits a specified amount to the account. This method should throw an Exception is the amount is negative.