

**Lakehead University**  
**Department of Computer Science**  
**CS 2477 Object Oriented Programming**  
**Assignment # 2**  
**Instructor: Dr. Jinan Fiaidhi**

Q1: How many guesses does it take to guess a secret number between 1 and N? For example, I'm thinking of a number between 1 and 100. I'll tell you whether your guess is too high or too low. Obviously, an intelligent first guess would be 50. If that's too low, an intelligent second guess would be 75. And so on. If we continue to divide the range in half, we'll eventually get down to one number. Because you can divide 100 seven times (50, 25, 12, 6, 3, 1, 0), it will take at most seven guesses to guess a number between 1 and 100. Write a Java application that lets the user input a positive integer, N, and then reports how many guesses it would take to guess a number between 1 and N.

Q2: A list is a sequential data structure. Design a List class that uses a comma-delimited String such as, "a,b,c,d,12,dog" to implement a list. Implement the following methods for this class:

```
void addItem( Object o ); // Use Object.toString()
String getItem(int position);
String toString();
void deleteItem(int position);
void deleteItem(String item);
int getPosition(String item);
String getHead(); // First element
List getTail(); // All but the first element
int length(); // Number of items
```

Q3: Define the MyRectangle2D class that contains:

- Two double data fields named x and y that specify the center of the rectangle with getter and setter methods. (Assume that the rectangle sides are parallel to x- or y- axes.)
- The data fields width and height with getter and setter methods.
- A no-arg constructor that creates a default rectangle with (0, 0) for (x, y) and 1 for both width and height.
- A constructor that creates a rectangle with the specified x, y, width, and height.
- A method getArea() that returns the area of the rectangle.
- A method getPerimeter() that returns the perimeter of the rectangle.
- A method contains(double x, double y) that returns true if the specified point (x, y) is inside this rectangle (see F I G U R E 1 (A)).
- A method contains(MyRectangle2D r) that returns true if the specified rectangle is inside this rectangle (see F I G U R E 1 (b)).
- A method overlaps(MyRectangle2D r) that returns true if the specified rectangle overlaps with this rectangle. (see F I G U R E 1 (c)).

Write a test program that creates a MyRectangle2D object r1 (new MyRectangle2D(2, 2, 5.5, 4.9)), displays its area and perimeter, and displays the result of r1.contains(3, 3), r1.contains(new MyRectangle2D(4, 5, 10.5, 3.2)), and r1.overlaps(new MyRectangle2D(3, 5, 2.3, 5.4)).

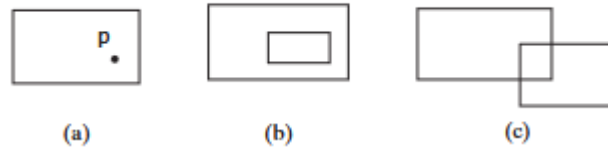


FIGURE 1 (A) point is inside the rectangle. (b) A rectangle is inside another rectangle. (c) A rectangle overlaps another rectangle.

Q4: Design a class named Account that contains:

- A private int data field named id for the account (default 0).
- A private double data field named balance for the account (default 0).
- A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.
- A private Date data field named dateCreated that stores the date when the account was created.
- A no-arg constructor that creates a default account.
- A constructor that creates an account with the specified id and initial balance.
- The accessor and mutator methods for id, balance, and annualInterestRate.
- The accessor method for dateCreated.
- A method named getMonthlyInterestRate() that returns the monthly interest rate.
- A method named getMonthlyInterest() that returns the monthly interest.
- A method named withdraw that withdraws a specified amount from the account.
- A method named deposit that deposits a specified amount to the account.

Use the Account class created to simulate an ATM machine. Create ten accounts in an array with id 0, 1, . . . , 9, and initial balance \$100. The system prompts the user to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, the main menu is displayed as shown in the sample run. You can enter a choice 1 for viewing the current balance, 2 for withdrawing money, 3 for depositing money, and 4 for exiting the main menu. Once you exit, the system will prompt for an id again. Thus, once the system starts, it will not stop.

```
Enter an id: 4 ↵Enter
Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 ↵Enter
The balance is 100.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 2 ↵Enter
Enter an amount to withdraw: 3 ↵Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 ↵Enter
The balance is 97.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 3 ↵Enter
Enter an amount to deposit: 10 ↵Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 ↵Enter
The balance is 107.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 4 ↵Enter

Enter an id:
```

**Due date: February 9.**