Concordia University
Comp 249 - Winter 2015
Object Oriented programming II
Assignment 2

Deadline:	By 11:59pm, Friday February 13, 2015
Evaluation:	3% of your final grade
Late Submission:	No late submission.
Teams:	The assignment can be done individually or in teams of 2 (from the same lecture section). Submit only one assignment per team.
Purpose:	The purpose of this assignment is to apply in practice the notions of inheritance, overriding, and access rights.

- **Problem specification.**

Illustration of a European roulette wheel

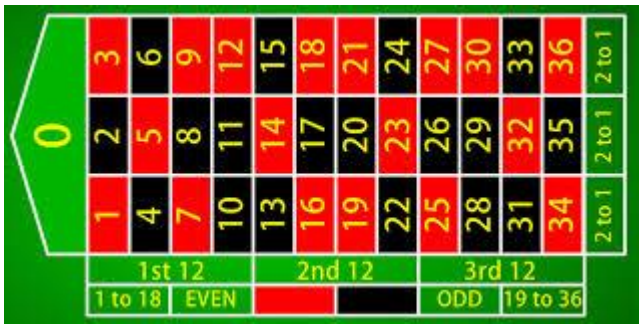

Illustration of a European roulette table layout

The game of roulette is one of the most popular casino games. It originated in Italy in the early 17th century but was developed and popularized in the 18th century, mainly in France.

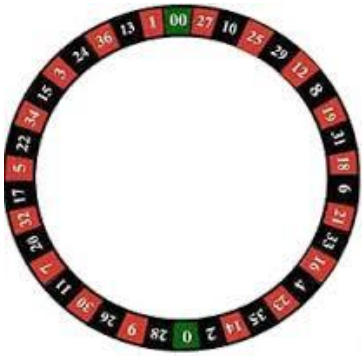

Illustration of an American roulette wheel

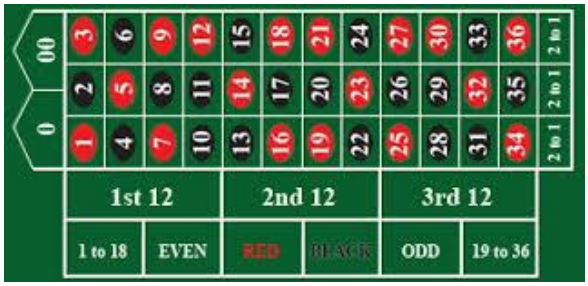

Illustration of a European roulette table layout

The players (usually a maximum of 6) sit around a table that consists of a table layout to place the bets and a numbered wheel. The dealer turns the wheel and launches a ball around it, and when the ball loses its velocity it falls on one of the 37 (38 for American roulette) numbers on the wheel.

The players that placed their tokens on the winning number get rewarded a multiple of their bets. Players can bet on the inside and on the outside of the table layout. Inside bets can be on a single (straight) number (pays 35 times), on two (split) numbers (pays 17 times), on three (street) numbers (pays 11 times), on four (square) numbers (pays 8 times), on five (top line – only in American roulette) numbers (pays 6 times) and on six (double street) numbers (pays 5 times). Outside bets on even, odd, red, black, first 18 (1-18), last 18 (19-36) numbers pay 1 time. Outside bets on first 12 (1-12), second 12 (13-24), third 12 (25-36), first column (1, 4,…, 34), second column (2, 5,…, 35), and third column numbers (3, 6,…, 36) pay 2 times.

- **Implementation.**

For simplicity reason you will implement only the basic rules of the game of roulette. Each class must implement the ***toString( )*** method to display the data member of an object using the ***System.println(…)*** method, and ***equals(…)*** method to compare two objects.

- The roulette table.

  The wheel numbers will be represented by a one-dimension array initialized with 0 as the first element and 26 as the last element for the European roulette. The sequence starts with 0 and ends with 2 for the American roulette.
  The spinning of the wheel is simulated using the ***Java.util.Random*** class to generate a random number representing the winning number.
  At the beginning of the program (when the table opens), the wheel map must be displayed on the screen using the ***System.println(…)*** method so that the players can start betting.

  The table layout is represented by the bets of each player betting a single token on a maximum of 5 different straight numbers. The players bet by inputting up to 5 numbers on the keyboard.
  The total bets and the total winnings (i.e. payouts) must be recorded for the table.

- The players.

  There are 6 players around the table and each player can bet on a maximum of 5 numbers, 1 token per number. There are two categories of

players, a regular player and a vip player. The regular player uses a token value $5 and the vip player has a token value $10. In addition, the vip player has rewards of 5% for the total bets and 10% for the total winnings. Also, vip players place their bets first on the table. You should at least have one vip player.
The total bets and the total winnings must be recorded for each player.

- The dealer.

  The dealer is responsible to turn the wheel, to calculate winnings and to make announcements.

- The driver code.

  At program start up, the program either creates a European roulette object or an American roulette object. The rules are essentially the same for European roulette and American roulette except for the wheel sequence numbers and the 00 on the American roulette table layout.
  The game ends after a minimal of 10 rolls of the wheel or after a timeout when the dealer decides to close the table.

- **Sample test cases.**

>Concordia European roulette v1.0 January 26 2015
>Wheel : 0-32-15-19-4-21-2-25-17-34-6-27-13-36-11-30-8-23-10-5-24-16-33-1-20-14-31-9-
22-18-29-7-28-12-35-3-26
>Dealer…Place your bets
>Player 1 : 8 12 26 23 7
>Player 2 : 32 6 0 17
>Player 3 : 14
>Player 4 : 8 23 11
>Player 5 : 18 35 21 10 22
>Player 6 : 19 1 33 26
>Dealer…No more bets
>Dealer…The winning number is 23
>Dealer…Player 1 wins $350, player 4 wins $175

- **Evaluation.**

  You will be evaluated mostly on the implementation of the classes, the relationship (is-a vs has-a) between the classes, the method overriding and the access rights of the class members.

- Evaluation criteria

Criteria	Marks
UML class diagram describing the classes and the class relationships, based on problem specification.	15%
Programming style, Comments and Javadoc.	15%
Implementation of class methods (constructor, accessor, mutator, helping, static), method overriding, access rights, etc.	40%
Driver code (main class).	10%
Output test cases (correctness and format).	20%

- **Required documents.**
  - UML diagram.
  - Source codes in Java.
  - Javadoc file in HTML.
  - Test cases for at least 10 rolls of the wheel.

- **Submission.**
  - Create one zip file, containing the necessary files (.java,.html and test cases). If the assignment is done individually, your file should be called *a2_studentID*, where a2 is the number of the assignment and *studentID* is your student ID number. If the work is done in a team of 2 people, the zip file should be called *a2_studentID1_studentID2* where *studentID1* and *studentID2* are the student ID numbers of each student.
  - Upload your zip file on eas (electronic assignment submission) at the URL: https://fis.encs.concordia.ca/eas/ as *Programming Assignment 2* before midnight on the due date.