## CSCI 145 PA \_\_6\_\_ Submission

## Due Date:\_\_\_April 10, 2023\_\_\_ Late (date and time):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Name(s):\_\_\_\_\_Ivan Leung\_\_\_\_\_\_ & \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Exercise 1 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa6;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 03 2023

Description: Play Rock, Paper, Scissors between the user and the computer

I certify that the code below is my own work.

Exception(s): N/A

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Rock.java

//

//Play Rock, Paper, Scissors with the user

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**import** java.util.Scanner;

**import** java.util.Random;

**public** **class** Rock {

**public** **static** **void** main(String[] args)

{

String personPlay; //User's play -- "R", "P", or "S"

String computerPlay; //Computer's play -- "R", "P", or "S"

**int** computerInt; //Randomly generated number used to determine

//computer's play

Scanner scan = **new** Scanner(System.***in***);

Random generator = **new** Random();

//Get player's play -- note that this is stored as a string

System.***out***.println("Enter your play: R, P, S");

personPlay = scan.next();

scan.close();

//Make player's play uppercase for ease of comparison

personPlay = personPlay.toUpperCase();

//Generate computer's play (0,1,2)

computerInt = generator.nextInt(3);

//Translate computer's randomly generated play to string

**switch** (computerInt)

{

**case** 1:

computerPlay = "R";

**break**;

**case** 2:

computerPlay = "P";

**break**;

**default**:

computerPlay = "S";

**break**;

}

//Print computer's play

System.***out***.println("Computer play is " + computerPlay);

//See who won. Use nested ifs instead of &&.

**if** (personPlay.equals(computerPlay))

System.***out***.println("It's a tie!");

**else** **if** (personPlay.equals("R"))

**if** (computerPlay.equals("S"))

System.***out***.println("Rock crushes scissors. You win!!");

**else**

System.***out***.println("Paper covers rock. You lose!!");//... Fill in rest of code

**else** **if** (personPlay.equals("P"))

**if** (computerPlay.equals("R"))

System.***out***.println("Paper covers rock. You win!!");

**else**

System.***out***.println("Scissors cuts paper. You lose!!");

**else**

**if** (computerPlay.equals("P"))

System.***out***.println("Scissors cuts paper. You win!!");

**else**

System.***out***.println("Rock crushes scissors. You lose!!");

}

}

Input/output below:

Enter your play: R, P, S

R

Computer play is R

It's a tie!

Enter your play: R, P, S

p

Computer play is R

Paper covers rock. You win!!

Enter your play: R, P, S

S

Computer play is P

Scissors cuts paper. You win!!

Enter your play: R, P, S

R

Computer play is S

Rock crushes scissors. You win!!

Enter your play: R, P, S

p

Computer play is S

Scissors cuts paper. You lose!!

Exercise 2 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa6;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 03 2023

Description: Determines the result of the elections in various formats

I certify that the code below is my own work.

Exception(s): N/A

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Election.java

//

//This file contains a program that tallies the results of

//an election. It reads in the number of votes for each of

//two candidates in each of several precincts. It determines

//the total number of votes received by each candidate, the

//percent of votes received by each candidate, the number of

//precincts each candidate carries, and the

//maximum winning margin in a precinct.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**import** java.util.Scanner;

**import** java.text.NumberFormat;

**public** **class** Election {

**public** **static** **void** main(String[] args) {

**int** votesForPolly; // number of votes for Polly in each precinct

**int** votesForErnest; // number of votes for Ernest in each precinct

**int** totalPolly; // running total of votes for Polly

**int** totalErnest; // running total of votes for Ernest

**int** wonByPolly;

**int** wonByErnest;

**int** ties;

String response; // answer (y or n) to the "more precincts" question

Scanner scan = **new** Scanner(System.***in***);

NumberFormat percent = NumberFormat.*getPercentInstance*();

percent.setMinimumFractionDigits(1);

System.***out***.println();

System.***out***.println("Election Day Vote Counting Program");

System.***out***.println();

// Initializations

totalPolly = 0;

totalErnest = 0;

wonByPolly = 0;

wonByErnest = 0;

ties = 0;

// Loop to "process" the votes in each precinct

**do** {

System.***out***.print("Vote for Polly: ");

votesForPolly = scan.nextInt();

System.***out***.print("Vote for Ernest: ");

votesForErnest = scan.nextInt();

totalPolly += votesForPolly;

totalErnest += votesForErnest;

**if** (votesForPolly == votesForErnest)

++ties;

**else** **if** (votesForPolly > votesForErnest)

++wonByPolly;

**else**

++wonByErnest;

System.***out***.print("More precincts? ");

response = scan.next();

} **while** (response.equalsIgnoreCase("Y"));

// Print out the results

scan.close();

System.***out***.println("Total votes for Polly: " + totalPolly + " (" + percent.format((**double**) totalPolly / (totalPolly + totalErnest)) + ")");

System.***out***.println("Total votes for Ernest: " + totalErnest + " (" + percent.format((**double**) totalErnest / (totalPolly + totalErnest)) + ")");

System.***out***.println("Precincts won by Polly: " + wonByPolly);

System.***out***.println("Precincts won by Ernest: " + wonByErnest);

System.***out***.println("Precincts tie: " + ties);

}

}

Input/output below:

Election Day Vote Counting Program

Vote for Polly: 8

Vote for Ernest: 4

More precincts? y

Vote for Polly: 7

Vote for Ernest: 10

More precincts? Y

Vote for Polly: 8

Vote for Ernest: 8

More precincts? n

Total votes for Polly: 23 (51.1%)

Total votes for Ernest: 22 (48.9%)

Precincts won by Polly: 1

Precincts won by Ernest: 1

Precincts tie: 1

Exercise 3 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa6;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 03 2023

Description: Determines the result of the elections in various formats

I certify that the code below is my own work.

Exception(s): N/A

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Temps.java

//

//This program reads in a sequence of hourly temperature

//readings (beginning with midnight) and prints the maximum

//temperature (along with the hour, on a 24-hour clock, it

//occurred) and the minimum temperature (along with the hour

//it occurred).

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**import** java.util.Scanner;

**public** **class** Temps {

// --------------------------------------------------

// Reads in a sequence of temperatures and finds the

// maximum and minimum read in.

// --------------------------------------------------

**public** **static** **void** main(String[] args) {

**final** **int** HOURS\_PER\_DAY = 8;

**int** temp; // a temperature reading

**int** maxTemp = -1000;

**int** timeOfMax = 0;

**int** minTemp = 1000;

**int** timeOfMin = 0;

Scanner scan = **new** Scanner(System.***in***);

// print program heading

System.***out***.println();

System.***out***.println("Temperature Readings for 8 Hour Period");

System.***out***.println();

**for** (**int** hour = 0; hour < HOURS\_PER\_DAY; hour++) {

System.***out***.print("Enter the temperature reading at " + hour + " hours: ");

temp = scan.nextInt();

**if** (temp > maxTemp) {

maxTemp = temp;

timeOfMax = hour;

}

**if** (temp < minTemp) {

minTemp = temp;

timeOfMin = hour;

}

}

scan.close();

// Print the results

System.***out***.println("Maximum temperature is " + maxTemp + " and occurred at hour " + timeOfMax);

System.***out***.println("Minimum tempreature is " + minTemp + " and occurred at hour " + timeOfMin);

}

}

Input/output below:

Temperature Readings for 8 Hour Period

Enter the temperature reading at 0 hours: 75

Enter the temperature reading at 1 hours: 78

Enter the temperature reading at 2 hours: 77

Enter the temperature reading at 3 hours: 80

Enter the temperature reading at 4 hours: 85

Enter the temperature reading at 5 hours: 83

Enter the temperature reading at 6 hours: 76

Enter the temperature reading at 7 hours: 74

Maximum temperature is 85 and occurred at hour 4

Minimum tempreature is 74 and occurred at hour 7

*Add more exercises as needed*

Exercise 4 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa6;

//Class Roulette for CSCI 145 PA 4 Spring 2023

//Modified by: Ivan Leung

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 04 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**import** java.util.\*;

//Class Roulette represents a roulette betting game.

**class** Roulette {

// public name constants -- accessible to others

**public** **final** **static** **int** ***BLACK*** = 0; // even numbers

**public** **final** **static** **int** ***RED*** = 1; // odd numbers

**public** **final** **static** **int** ***GREEN*** = 2; // 00 OR 0

**public** **final** **static** **int** ***NUMBER*** = 3; // number bet

**public** **final** **static** **int** ***MIN\_NUM*** = 1; // smallest number to bet

**public** **final** **static** **int** ***MAX\_NUM*** = 36; // largest number to bet

**public** **final** **static** **int** ***MIN\_BET*** = 1; // minimum amount to bet

// private name constants -- internal use only

**private** **final** **static** **int** ***MAX\_POSITIONS*** = ***MAX\_NUM*** + 2; // number of positions on wheel

**private** **final** **static** **int** ***NUMBER\_PAYOFF*** = ***MAX\_NUM*** - 1; // payoff for number bet

**private** **final** **static** **int** ***COLOR\_PAYOFF*** = 2; // payoff for color bet

// private variables -- internal use only

**private** **static** **int** *ballPosition* = 0; // 00, 0, 1 .. MAX\_NUM

**private** **static** **int** *color* = ***GREEN***; // GREEN, RED, OR BLACK

// private variables -- testing only

// private static int next = 0; // next value in the list

// private static int[] randValues = { 20, 5, 0, 1, 36 }; // 5 values

// Contains the main processing loop for the roulette game.

**public** **static** **void** main(String[] args) {

Scanner scan = **new** Scanner(System.***in***);

Player player = **new** Player("Jane", 100); // $100 to start for Jane

**boolean** done = **false**;

// int currentSpin;

**int** initialAmount = player.getMoney();

**int** totalPayment;

System.***out***.println("Author: Ivan Leung\n");

*welcomeMessage*();

**while** (!done) {

System.***out***.println("Money available for " + player.getName() + ": " + player.getMoney());

*betOptions*();

// Add code so player can make a bet

player.makeBet(scan);

System.***out***.println();

// spin() and display value

// currentSpin = spin();

*spin*();

System.***out***.println();

// Determines payment

player.payment(*payoff*(player.getBet(), player.getBetType(), player.getNumber()));

done = !player.playAgain(scan);

System.***out***.println();

}

totalPayment = player.getMoney() - initialAmount;

System.***out***.println(player.getName() + (totalPayment >= 0 ? " won " : " lost ") + (**int**) Math.*abs*((**double**) totalPayment) + " for this game.");

System.***out***.println("Game over! Thanks for playing.");

scan.close();

}

// =====================================================================

// Presents welcome message

// =====================================================================

**public** **static** **void** welcomeMessage() {

System.***out***.println("Welcome to a simple version of roulette game.");

System.***out***.println("You can place a bet on black, red, or a number.");

System.***out***.println("A color bet is paid " + ***COLOR\_PAYOFF*** + " times the bet amount.");

System.***out***.println("A number bet is paid " + ***NUMBER\_PAYOFF*** + " times the bet amount.");

System.***out***.println("You can bet on a number from " + ***MIN\_NUM*** + " to " + ***MAX\_NUM*** + ".");

System.***out***.println("Gamble responsibly. Have fun and good luck!\n");

}

// =====================================================================

// Presents betting options

// =====================================================================

**public** **static** **void** betOptions() {

System.***out***.println("Betting Options:");

System.***out***.println(" 1. Bet on black (even numbers)");

System.***out***.println(" 2. Bet on red (odd numbers)");

System.***out***.println(" 3. Bet on a number between " + ***MIN\_NUM*** + " and " + ***MAX\_NUM***);

System.***out***.println();

}

// Spins the wheel

**public** **static** **int** spin() {

**int** result;

// use nextRandom() for testing now

result = *nextRandom*();

Roulette.*ballPosition* = result;

**if** (result == 0 || result == 37)

Roulette.*color* = Roulette.***GREEN***;

**else** **if** ((result & 1) == 0)

Roulette.*color* = Roulette.***BLACK***;

**else**

Roulette.*color* = Roulette.***RED***;

System.***out***.println("Spinning ...");

System.***out***.print("Current number: ");

**if** (result == 37)

System.***out***.print("00, color: ");

**else**

System.***out***.print(result + ", color: ");

**if** (Roulette.*color* == Roulette.***GREEN***)

System.***out***.println("Green");

**else** **if** (Roulette.*color* == Roulette.***BLACK***)

System.***out***.println("Black");

**else**

System.***out***.println("Red");

**return** result;

// comment above code and add your code to spin

}

// Payoff method for number bet

**public** **static** **int** payoff(**int** betAmt, **int** betType, **int** numberBet) {

**int** pay = 0;

**if** (Roulette.*ballPosition* == 0 || Roulette.*ballPosition* == 37) {

**return** pay;

}

**else** **if** (betType == 1) {

**if** (Roulette.*color* == Roulette.***BLACK***)

**return** pay = betAmt \* Roulette.***COLOR\_PAYOFF***;

}

**else** **if** (betType == 2) {

**if** (Roulette.*color* == Roulette.***RED***)

**return** pay = betAmt \* Roulette.***COLOR\_PAYOFF***;

}

**else** {

**if** (numberBet == Roulette.*ballPosition*)

**return** pay = betAmt \* Roulette.***NUMBER\_PAYOFF***;

}

**return** pay;

}

// Returns a simulated "random" value for testing

// Assume a value between 0 and 36

**public** **static** **int** nextRandom() {

// int num = randValues[next];

// next++;

// next = next % randValues.length; // back to 0 if needed

Random rand = **new** Random();

**int** num = rand.nextInt(Roulette.***MAX\_POSITIONS***);

**return** num;

}

}

**package** pa6;

//Class Player for CSCI 145 PA 4 Spring 2023

//Modified by: Ivan Leung

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 04 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**import** java.util.\*;

//Class Player represents one roulette player.

**class** Player {

**private** **static** **final** **int** ***RELOAD\_AMOUNT*** = 100;

**private** **int** bet, money, betType, number;

**private** String name;

// The Player constructor sets up name and initial available money.

**public** Player(String playerName, **int** initialMoney) {

name = playerName;

money = initialMoney;

}

// Returns this player's name.

**public** String getName() {

**return** name;

}

// Returns this player's current available money.

**public** **int** getMoney() {

**return** money;

}

**public** **int** getBetType() {

**return** betType;

}

**public** **int** getBet() {

**return** bet;

}

**public** **int** getNumber() {

**return** number;

}

// Prompts the user and reads betting information.

**public** **void** makeBet(Scanner scan) {

System.***out***.print("Enter a bet option, " + name + " (1, 2, or 3): ");

betType = scan.nextInt();

**while** (betType < 1 || betType > 3) {

System.***out***.print("Invalid option (must be between 1 and 3).\nEnter a bet option, " + name + " (1, 2, or 3): ");

betType = scan.nextInt();

}

**if** (betType == 3) {

System.***out***.print("Enter a number: ");

number = scan.nextInt();

**while** (number < Roulette.***MIN\_NUM*** || number > Roulette.***MAX\_NUM***) {

System.***out***.print("Invalid number (must be between 1 and 36).\nEnter a number: ");

number = scan.nextInt();

}

}

System.***out***.print("How much to bet: ");

bet = scan.nextInt();

**while** (bet < Roulette.***MIN\_BET*** || bet > money) {

System.***out***.print("Invalid amount (must be at least 1).\nHow much to bet: ");

bet = scan.nextInt();

}

System.***out***.print("You chose to bet $" + bet + " on ");

**if** (betType == 1)

System.***out***.println("Black color");

**else** **if** (betType == 2)

System.***out***.println("Red color");

**else**

System.***out***.println("number " + number + ".");

money = money - bet;

}

// Determines if the player wants to play again.

**public** **boolean** playAgain(Scanner scan) {

String answer;

System.***out***.print("Play again, " + name + "? [y/n] ");

answer = scan.next();

**return** (answer.equals("y") || answer.equals("Y"));

}

// payment method (determines winnings)

**public** **int** payment(**int** pay) {

money += pay;

**if** (pay > 0)

System.***out***.println(name + " won " + pay + ".");

**else**

System.***out***.println(name + " lost this round.");

**return** pay;

}

}

Input/output below:

Author: Ivan Leung

Welcome to a simple version of roulette game.

You can place a bet on black, red, or a number.

A color bet is paid 2 times the bet amount.

A number bet is paid 35 times the bet amount.

You can bet on a number from 1 to 36.

Gamble responsibly. Have fun and good luck!

Money available for Jane: 100

Betting Options:

1. Bet on black (even numbers)

2. Bet on red (odd numbers)

3. Bet on a number between 1 and 36

Enter a bet option, Jane (1, 2, or 3): 4

Invalid option (must be between 1 and 3).

Enter a bet option, Jane (1, 2, or 3): 1

How much to bet: 10

You chose to bet $10 on Black color

Spinning ...

Current number: 1, color: Red

Jane lost this round.

Play again, Jane? [y/n] Y

Money available for Jane: 90

Betting Options:

1. Bet on black (even numbers)

2. Bet on red (odd numbers)

3. Bet on a number between 1 and 36

Enter a bet option, Jane (1, 2, or 3): 2

How much to bet: 10

You chose to bet $10 on Red color

Spinning ...

Current number: 7, color: Red

Jane won 20.

Play again, Jane? [y/n] y

Money available for Jane: 100

Betting Options:

1. Bet on black (even numbers)

2. Bet on red (odd numbers)

3. Bet on a number between 1 and 36

Enter a bet option, Jane (1, 2, or 3): 3

Enter a number: 38

Invalid number (must be between 1 and 36).

Enter a number: 18

How much to bet: 10

You chose to bet $10 on number 18.

Spinning ...

Current number: 1, color: Red

Jane lost this round.

Play again, Jane? [y/n] y

Money available for Jane: 90

Betting Options:

1. Bet on black (even numbers)

2. Bet on red (odd numbers)

3. Bet on a number between 1 and 36

Enter a bet option, Jane (1, 2, or 3): 2

How much to bet: 0

Invalid amount (must be at least 1).

How much to bet: 10

You chose to bet $10 on Red color

Spinning ...

Current number: 0, color: Green

Jane lost this round.

Play again, Jane? [y/n] y

Money available for Jane: 80

Betting Options:

1. Bet on black (even numbers)

2. Bet on red (odd numbers)

3. Bet on a number between 1 and 36

Enter a bet option, Jane (1, 2, or 3): 1

How much to bet: 10

You chose to bet $10 on Black color

Spinning ...

Current number: 9, color: Red

Jane lost this round.

Play again, Jane? [y/n] y

Money available for Jane: 70

Betting Options:

1. Bet on black (even numbers)

2. Bet on red (odd numbers)

3. Bet on a number between 1 and 36

Enter a bet option, Jane (1, 2, or 3): 1

How much to bet: 10

You chose to bet $10 on Black color

Spinning ...

Current number: 0, color: Green

Jane lost this round.

Play again, Jane? [y/n] y

Money available for Jane: 60

Betting Options:

1. Bet on black (even numbers)

2. Bet on red (odd numbers)

3. Bet on a number between 1 and 36

Enter a bet option, Jane (1, 2, or 3): 1

How much to bet: 20

You chose to bet $20 on Black color

Spinning ...

Current number: 6, color: Black

Jane won 40.

Play again, Jane? [y/n] n

Jane lost 20 for this game.

Game over! Thanks for playing.

Answer for Question 1

The conditional operator is a ternary operator which requires a condition and two results. The condition is placed before a question mark and the two results are placed after the question mark separated by a colon. If the condition is evaluated to be true, then the first result will be executed, otherwise the second result will be executed. Here is a basic format of a conditional operator below:

(condition ? true : false)

The conditional operator can be used wherever an if-else statement (exactly one if and one else) is being used because they are interchangeable. The conditional operator usually provide better readability because it can be written in one single line instead of four lines in if-else statement.

Answer for Question 2

Among the three loop, do-while loop is different from the other two. The main difference is that do-while loop statement is executed at least one time or more while for loop and while loop are executed at least zero times or more.

Extra Credit – provide if applicable

Pseudocode below if applicable:

Source code below:

**package** pa6;

//Class Player for CSCI 145 PA 4 Spring 2023

//Modified by: Ivan Leung

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 04 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**public** **class** Coin {

**public** **final** **int** HEADS = 0;

**public** **final** **int** TAILS = 1;

**private** **int** face;

// ---------------------------------------------

// Sets up the coin by flipping it initially.

// ---------------------------------------------

**public** Coin() {

flip();

}

// -----------------------------------------------

// Flips the coin by randomly choosing a face.

// -----------------------------------------------

**public** **void** flip() {

face = (**int**) (Math.*random*() \* 2);

}

// ---------------------------------------------------------

// Returns true if the current face of the coin is heads.

// ---------------------------------------------------------

**public** **boolean** isHeads() {

**return** (face == HEADS);

}

// ----------------------------------------------------

// Returns the current face of the coin as a string.

// ----------------------------------------------------

**public** String toString()

{

String faceName;

**if** (face == HEADS)

faceName = "Heads";

**else**

faceName = "Tails";

**return** faceName;

}

}

**package** pa6;

//Class Player for CSCI 145 PA 4 Spring 2023

//Modified by: Ivan Leung

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 04 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Runs.java

//

//Finds the length of the longest run of heads in 100 flips of a coin.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** Runs {

**public** **static** **void** main(String[] args) {

**final** **int** FLIPS = 100; // number of coin flips

**int** currentRun = 0; // length of the current run of HEADS

**int** maxRun = 0; // length of the maximum run so far

// Create a coin object

Coin coin = **new** Coin();

// Flip the coin FLIPS times

**for** (**int** i = 0; i < FLIPS; i++) {

// Flip the coin & print the result

coin.flip();

System.***out***.println(coin);

// Update the run information

**if** (coin.isHeads()) {

++currentRun;

}

**else** {

**if** (currentRun > maxRun) {

maxRun = currentRun;

}

currentRun = 0;

}

}

// Print the results

System.***out***.println("\nThe longest run of Head is " + maxRun + " Heads.");

}

}

Input/output below:  
  
Tails

Heads

Heads

Heads

Heads

Heads

Heads

Tails

Heads

Tails

Heads

Heads

Tails

Heads

Tails

Tails

Heads

Heads

Heads

Heads

Heads

Heads

Heads

Heads

Heads

Heads

Tails

Heads

Tails

Heads

Heads

Heads

Heads

Tails

Tails

Tails

Tails

Tails

Heads

Tails

Heads

Heads

Tails

Heads

Heads

Heads

Heads

Tails

Tails

Heads

Heads

Heads

Heads

Heads

Heads

Tails

Heads

Heads

Tails

Heads

Tails

Tails

Heads

Tails

Heads

Tails

Heads

Heads

Tails

Heads

Tails

Tails

Tails

Heads

Heads

Tails

Tails

Heads

Heads

Tails

Heads

Heads

Heads

Heads

Tails

Heads

Tails

Heads

Tails

Tails

Tails

Tails

Tails

Tails

Heads

Heads

Heads

Heads

Heads

Tails

The longest run of Head is 10 Heads.