Java Lab – Methods in Java Continued

IMPORTANT! Save all your work to a safe location such as oneDrive.

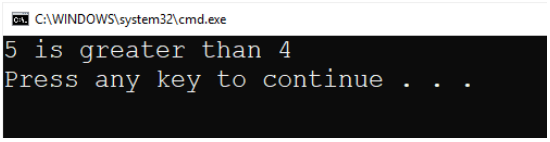
Create a folder for SDPD into which you will save all your work for this module, arranged how you wish. Ideally you should create a folder each week for your lab exercises. Note that you should create **a separate file** for each exercise.

Exercise 1

Create a program that uses a method called higher(). The method should output which number is the higher number among two integers provided to it as arguments.

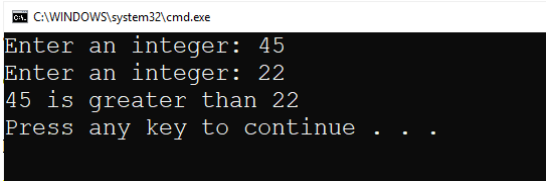
```
higher(5, 4);
```

Sample output:



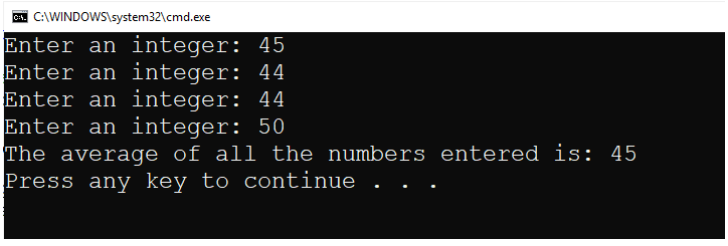
Exercise 2

Amend your code from exercise 1 so that the user is prompted to input the 2 numbers. The 2 prompts for input should be in the main method – there should be no need to change the method created in the previous exercise.



Exercise 3

Create a program that uses a method called average(). The method should output the average of 4 integers provided to it as arguments. Sample output:



Exercise 4

The distance a vehicle travels can be calculated as follows:

*Distance = Speed * Time*

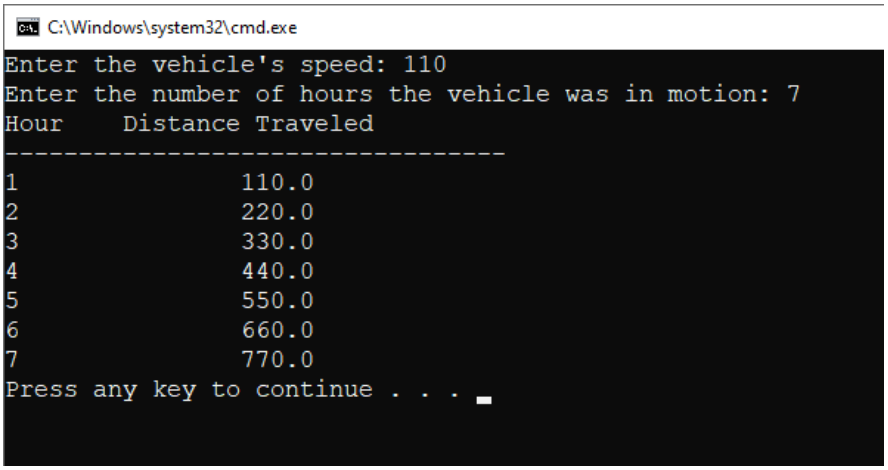
Write a method named *distance* that accepts a vehicle's speed and time as arguments, andreturns the distance the vehicle has travelled. This is all that your method will do – calculate the distance based on the time and speed provided to it, and return that calculation.

Use this method to produce the program and output as shown below.

Your program should ensure that the speed must be greater than zero when entered, otherwise the user is asked to input again.

Your program should ensure that the number of hours in motion is at least 1, otherwise the user is asked to input again.

Sample Output:

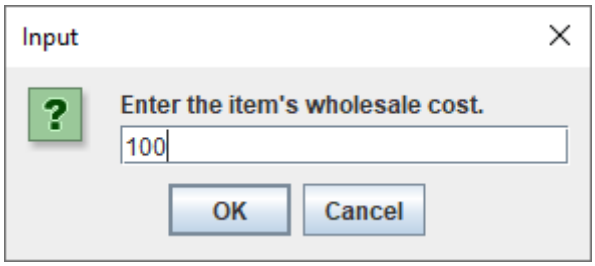


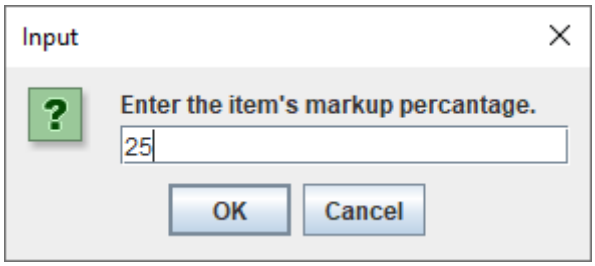
Exercise 5

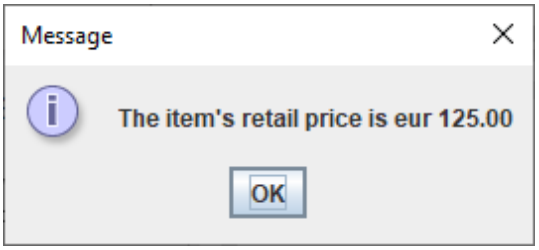
Write a program that asks the user to enter an item's wholesale cost and its markup percentage. It should then display the item's retail price. For example:

- If an item's wholesale cost is 5.00 and its markup percentage is 100 percent, then the item's retail price is 10.00.
- If an item's wholesale cost is 5.00 and its markup percentage is 50 percent, then the item's retail price is 7.50.

The program should have a method named calculateRetail() that receives the wholesale cost and the markup percentage as arguments, and returns the retail price of the item. Your output should be similar to as shown:







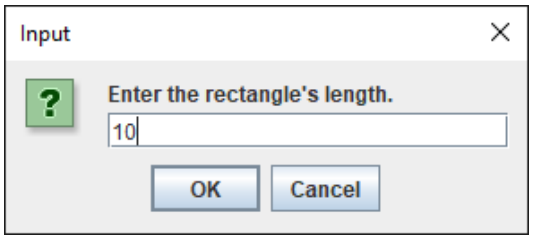
Exercise 6

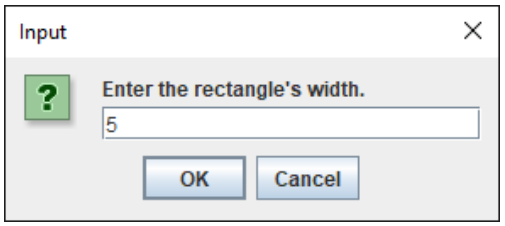
Create a program that will prompt the user to enter the width and length of a rectangle, and then display the rectangle's area. The program calls the following methods:

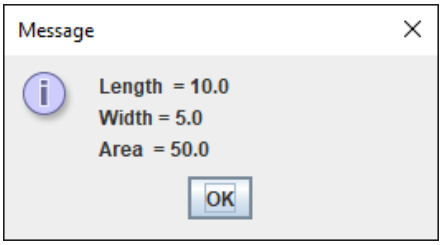
- getLength—This method should ask the user to enter the rectangle's length, and then return that value as a double.
- getWidth—This method should ask the user to enter the rectangle's width, and then return that value as a double.
- getArea—This method should accept the rectangle's length and width as arguments, and return the rectangle's area. The area is calculated by multiplying the length by the width.
- displayData—This method should accept the rectangle's length, width, and area as arguments, and display them in an appropriate message on the screen.

Your main method should look similar to as shown below. Create the above methods to ensure this works correctly.

```
public static void main(String[] args)
{
   double length;    // The rectangle's length
   double width;     // The rectangle's width
   double area;      // The rectangle's area

   // Get the rectangle's length from the user.
   length = getLength();

   // Get the rectangle's width from the user.
   width = getWidth();

   // Get the rectangle's area.
   area = getArea(length, width);

   // Display the rectangle data.
   displayData(length, width, area);

}
```





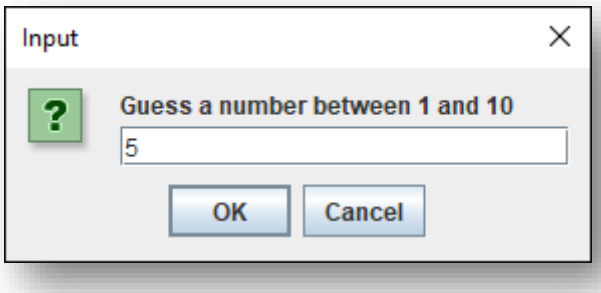


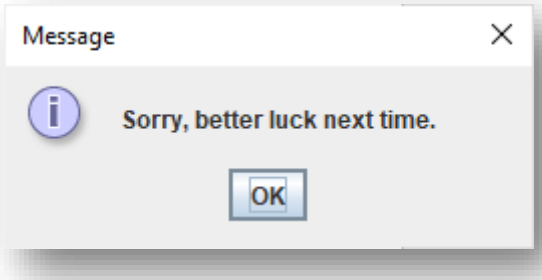
Exercise 7

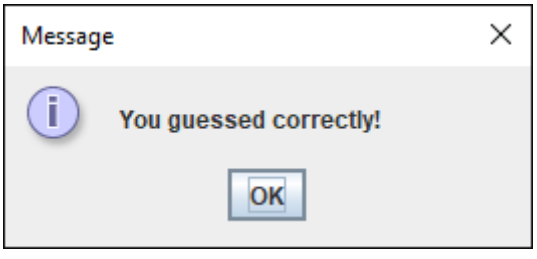
Create a guessing game where a Java program randomly selects a number between 1 and 10 (This ishidden from the user). The user then gets a chance to guess the number.

Construct your main method as shown below, and create the methods required to get this game to work similar to the screen shots as shown below:

```
public static void main(String[] args)
{
   int compNum;  // To hold random number between 1 and 10
   int guess;    // To hold the user's choice

   // Get the computer's random number between 1 and 10.
   compNum = computerRandom();

   // Get the user's guess.
   guess = userGuess();

   // Get the user's guess against the computers random number
   checkGuess(guess, compNum);
}
```







Exercise 8

Write a program that tests your ability to successfully guess a colour!
The program should randomly select the name of a colour from the following list of words:

Red, Green, Blue, Orange, Yellow

To select a colour, the program can generate a random number. For example, if the number is 0, the selected colour is *Red*; if the number is 1, the selected colour is *Green*; and so forth.

Next, the program should ask the user to guess the colour that the computer has selected.

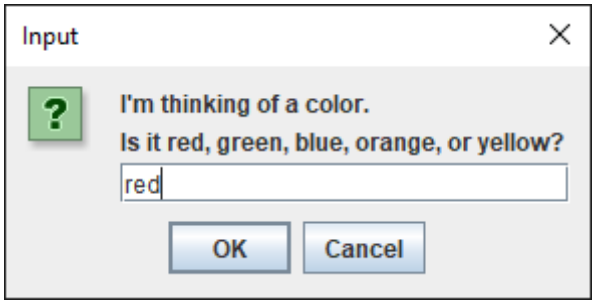
After the user has entered his or her guess, the program should display the name of the randomlyselected colour. The program should repeat this 5 times and then display the number of times the user correctly guessed the selected colour.

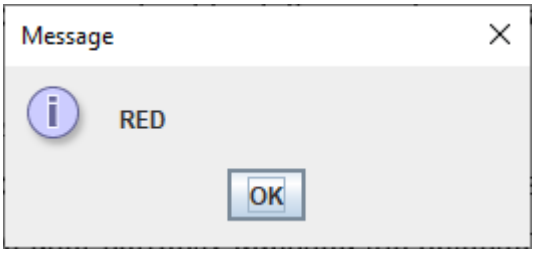
Be sure to modularize the program into methods that perform each major task.

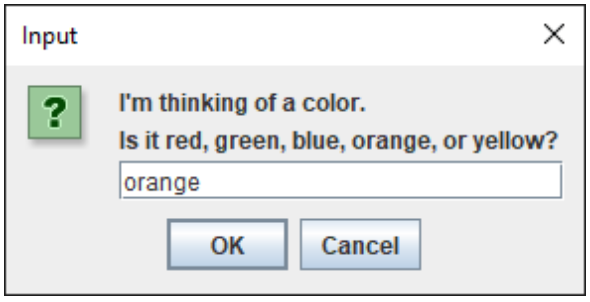
Your methods should include:

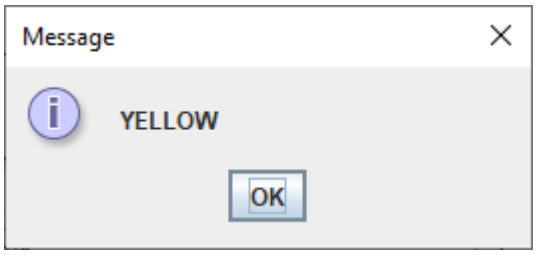
A method called computerChoice that will create the computers random colour choice, that will return either red, green, blue, orange or yellow.

A method called userChoice that will prompt the user to enter a choice of colour using JOptionPane and return the value entered.

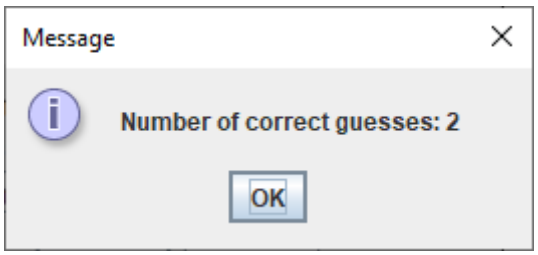








Result after program has completed running:



Exercise 9

Write a program that asks the user to **enter a distance in meters**. The program will then present the following menu of selections:

1. Convert to kilometers
2. Convert to inches
3. Convert to feet
4. Quit the program

The program will convert the distance to kilometers, inches, or feet, depending on the users selection. Here are the specific requirements:

The program will have 5 methods in total. One for each calculation, and a simple void method with no arguments that outputs the menu to the console.

Your main method should be as follows:

```
public static void main(String[] args)
{
   String input;         // To hold keyboard input
   int selection;        // Menu selection
   double distance;      // Distance in meters

   // Create a Scanner object for keyboard input.
   Scanner keyboard = new Scanner(System.in);

   // Get a distance.
   System.out.print("Enter a distance in meters: ");
   distance = keyboard.nextDouble();

   // Display the menu and process the user's
   // selection until 4 is selected.
   do
   {
      // Display the menu.
      menu();

      // Get the user's selection.
      System.out.print("\nEnter your choice: ");
      selection = keyboard.nextInt();

      // Validate the user's selection.
      while (selection < 1 || selection > 4)
      {
         System.out.print("Invalid selection. Enter your choice: ");
         selection = keyboard.nextInt();
      }

      // Process the user's selection.
      switch (selection)
      {
         case 1 : showKilometers(distance);
                  break;
         case 2 : showInches(distance);
                  break;
         case 3 : showFeet(distance);
                  break;
         case 4 : System.out.println("Bye!");
      }

      System.out.println();

   } while (selection != 4);
}
```

• Write a void method named showKilometers, which accepts the number of meters as an argument. The method should display the argument converted to kilometers. Convert meters to kilometers using the following formula:

kilometers = meters * 0.001

Write a void method named showInches, which accepts the number of meters as an argument. The method should display the argument converted to inches. Convert themeters to inches using the following formula:

inches = meters * 39.37

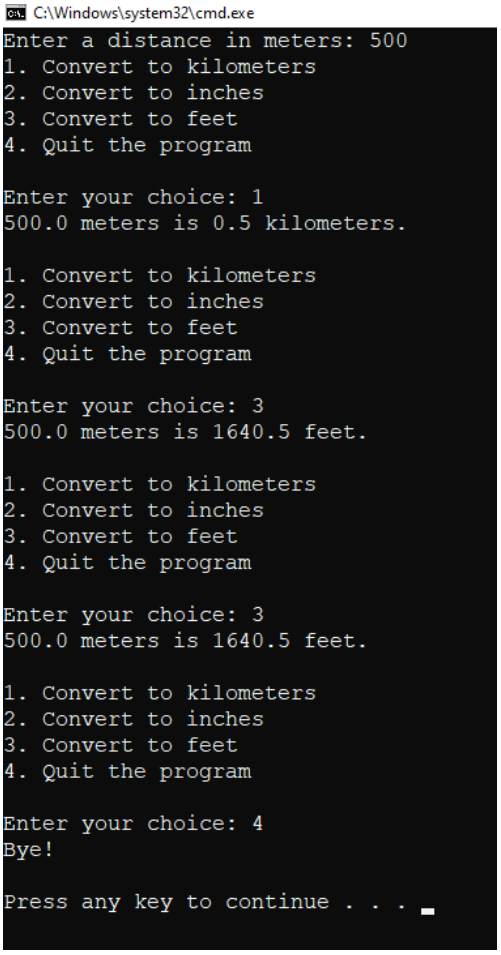
• Write a void method named showFeet, which accepts the number of meters as anargument. The method should display the argument converted to feet. Convert the meters to feet using the following formula:

feet = meters * 3.281

• Write a void method named menu that displays a menu of selections. This method should not accept any arguments.

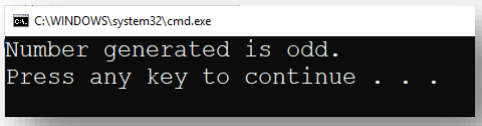
- The program should continue to display the menu until the user enters 4 to quit the program.
- The program should not accept negative numbers for the distance in meters.
- If the user selects an invalid choice from the menu, the program should display an error message.

Here is an example session with the program, using console input:

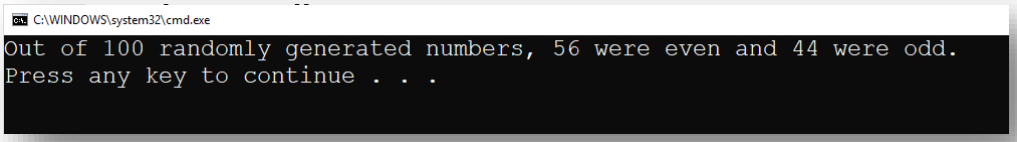


Exercise 10

Create a program in java that will determine if a number is odd or even. Your program should contain a method that returns a Boolean (either true or false) that determine whether a number is odd or even.



Amend your code so that the main method uses a loop to generate 100 random numbers and uses the method to check whether or not it is odd or even. Your results should be similar to as shown:



Exercise 11

Create a program that calls a void method without arguments called fileRead(). Your code should be similar to as shown below:

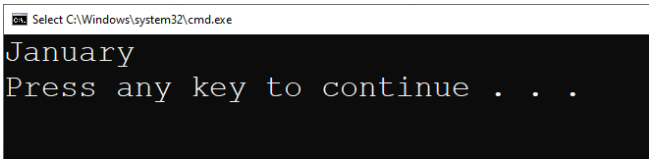
```
1  import java.util.Scanner;
2  import java.io.*;
3
4  public class methodsExceptions1
5  {
6      public static void main(String[] args) throws IOException
7      {
8
9          fileRead();
10
11     }
12
13 }
```

Create the fileRead() method so that the program will read the first line from a file called “months.txt” (File provided on moodle). Your method code may contain code similar to as shown below:

```
String lineRead;
File myFile = new File("months.txt");

Scanner inputFile = new Scanner(myFile);
lineRead = inputFile.nextLine();

System.out.println(lineRead);
```

Note that your *fileRead()* method will also need the `throws IOException` clause.

When your program executes, your output should be similar to as shown:



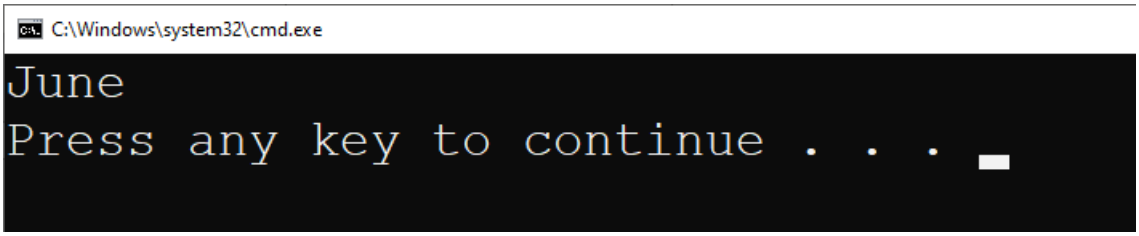
Amend your program with the following changes:

- The method should have a single integer parameter that specifies the line to be read from the file.

For example, calling the method as follows…

```
fileRead(6);
```

…would result in the following output:



- Next, change the method so that the line is returned as a string.

For example, calling the method as follows…

```
myvar = fileRead(6);
```

…would result in the variable `myvar` containing the contents of line 6 of the file.

Your method should return an error message if the number specified as an argument is greater than the number of lines in the file, eg:

```
fileRead(100);
```

This would return the following message as a string:

```
FILE READ ERROR: There are only 12 lines of text in this file
```

(This assumes that the file has 12 lines)

- Next, change the method so that the method can be called with 2 arguments, a number (representing the line number to read) and a String, f representing the file name to read from .For example:

```
myvar = fileRead(9, "months.txt");
```

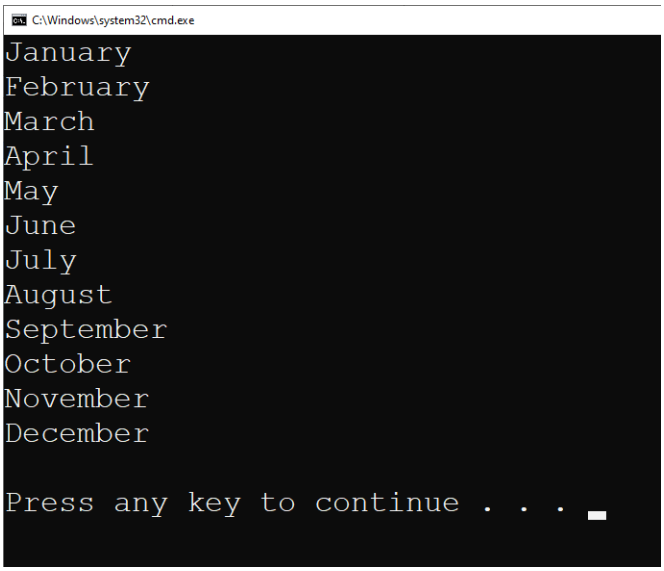
Would read the 9th line from the file *months.txt*

- Next, change the method so that if the method has the number 0 passed to it as the first argument, the entire contents of the file are returned

For example, calling the fileRead() method as shown…

```
myvar = fileRead(0, "months.txt");
System.out.println(myvar);
```

…would result in the following output:



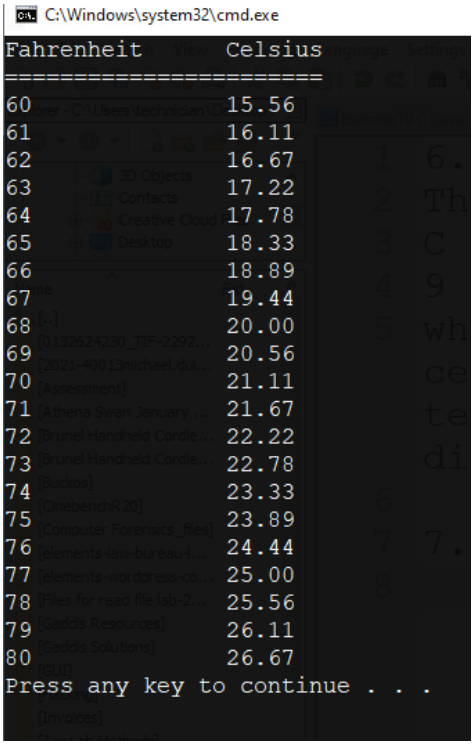
Exercise 12

The formula for converting a temperature from Fahrenheit to Celsius is:

```
0.55555 * (tempInFarenheit - 32)
```

where F is the Fahrenheit temperature and C is the Celsius temperature.

Write a method named celsius() that accepts a Fahrenheit temperature as an argument. The method should return the temperature, converted to Celsius. Demonstrate the method by calling it in a loop thatdisplays a table of the Fahrenheit temperatures 0 through 20 and their Celsius equivalents.



Exercise 13

Write a program that lets the user play the game of Rock, Paper, Scissors against the computer. The program should work as follows.

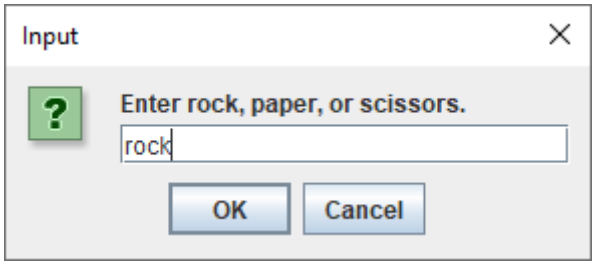
1. When the program begins, a random number in the range of 1 through 3 is generated. If the number is 1, then the computer has chosen rock. If the number is 2, then the computer has chosen paper. If the number is 3, then the computer has chosen scissors. (Don't display the computer's choice yet.)
2. The user enters his or her choice of "rock", "paper", or "scissors" at the keyboard. (You can use amenu if you prefer.)
3. The computer's choice is displayed.
4. A winner is selected according to the following rules:

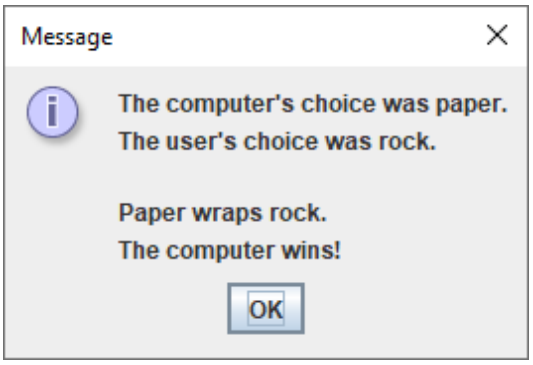
- If one player chooses rock and the other player chooses scissors, then rock wins. (The rocksmashes the scissors.)
- If one player chooses scissors and the other player chooses paper, then scissors wins.(Scissors cuts paper.)
- If one player chooses paper and the other player chooses rock, then paper wins. (Paper wrapsrock.)
- If both players make the same choice, the game must be played again to determine the winner.Be sure to divide the program into methods that perform each major task.

Your methods should include:

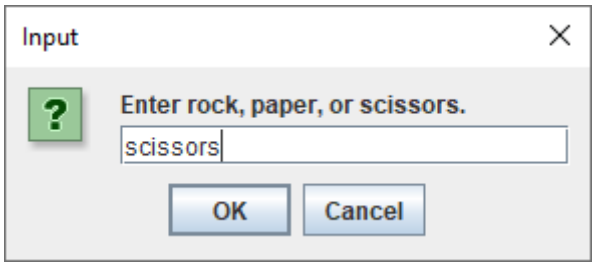
- computerChoice – this will randomly choose rock, paper or scissors, and return this as a string.
- userChoice – this will prompt the user to enter their choice or rock, paper or scissors, and return their choice as a string.
- determineWinner – the code for this **is provided** on moodle. Download this method and add it to your solution.

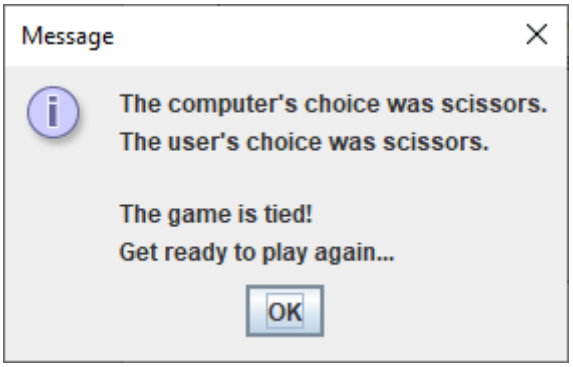
Your output should be similar to as shown below:



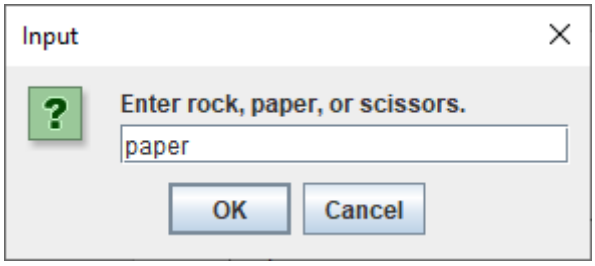


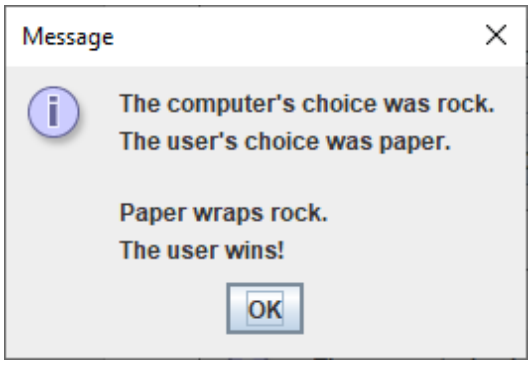
or


Input
Enter rock, paper, or scissors.
scissors
OK
Cancel


Message
The computer's choice was scissors.
The user's choice was scissors.
The game is tied!
Get ready to play again...
OK

or


Input
Enter rock, paper, or scissors.
paper
OK
Cancel


Message
The computer's choice was rock.
The user's choice was paper.
Paper wraps rock.
The user wins!
OK

Exercise 14

Write a program that in java with a method that counts the number of vowels in a string. The method should accept a string as a parameter and return an integer containing the number of vowels.

Hint:
You can use charAt for this

```
if (myVar.charAt(i) == 'a')
```