Java Lab – Methods in Java

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

Goal: Create a program in Java with a void method called *displayMessage* that outputs a welcomemessage to the console using the method.

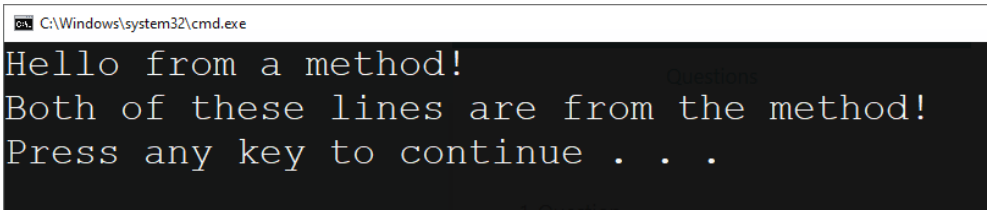
The code below shows how to create a simple *void* method that outputs a message. Note that the method is from lines 9 to 12, and the method is called from inside the main method, on line 6.

```
1  public class BasicMethod1
2  {
3      public static void main(String[] args)
4      {
5
6          displayMessage();
7      }
8
9      public static void displayMessage()
10     {
11         System.out.println("Hello from a method!");
12     }
13 }
```

Exercise 2

Goal: Create a program in Java with a void method.

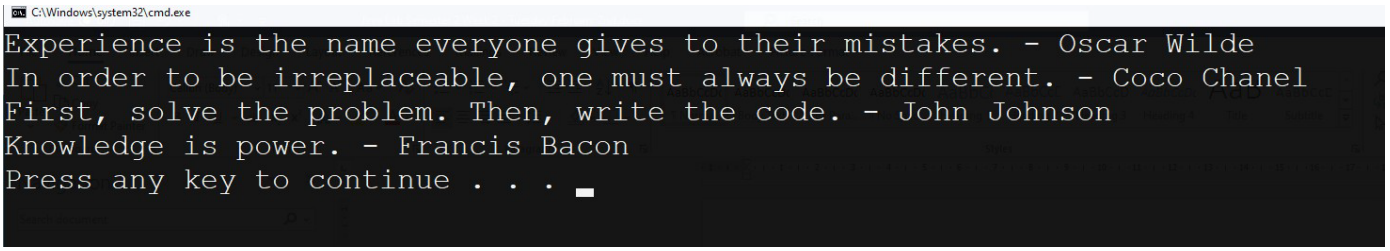
Your program should contain a void method called **welcome()** that, when called, outputs 2 lines to the console with the text as shown:



Exercise 3

voidMethods.java *

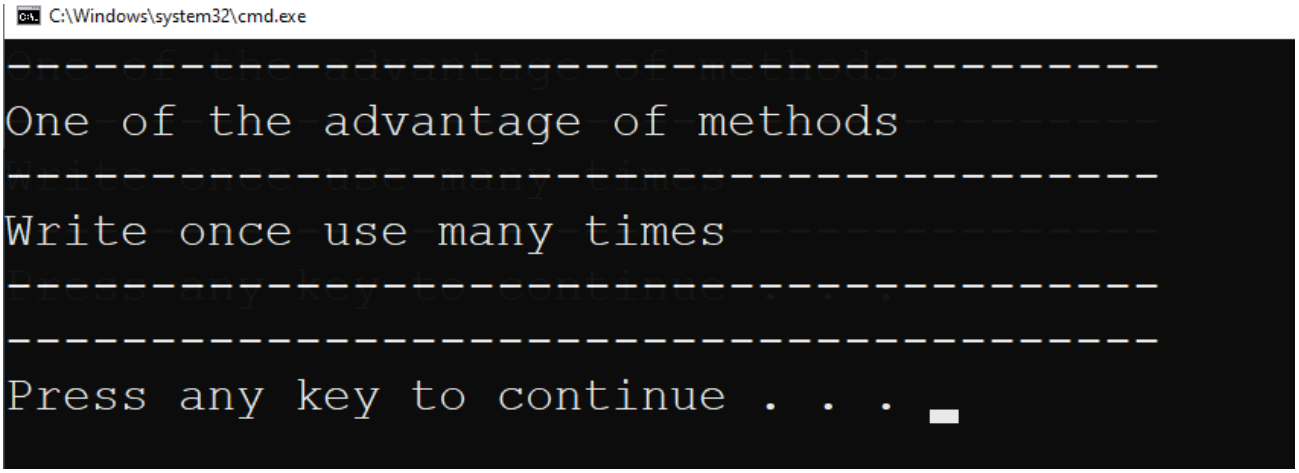
```java
public class voidMethods
{
    public static void main(String[] args)
    {
        oscar();
        coco();
        john();
        francis();
    }

}
```



Exercise 4

```
1  /**
2   *   This program defines and calls a method.
3   */
4  public class outputLine
5  {
6     public static void main(String[] args)
7     {
8        dashLine();
9        System.out.println("One of the advantage of methods");
10       dashLine();
11       System.out.println("Write once use many times");
12       dashLine();
13       dashLine();
14    }
15
16    /**
17     *      The dashLine method displays a line with 40 dashes.
18     */
19
20 }
```

Consider the code shown above. Write the void method required to produce the output as shown here:

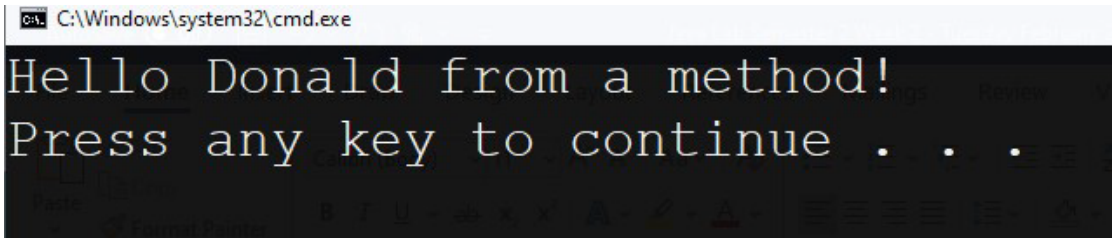


Exercise 5

Write a program with a void method named ***greetings()*** that accepts a String as an argument and displays a greeting message. Call this method from main().

```
1  /**
2   *   This program defines and calls a method with one String parameter
3   */
4
5  public class greetMethod
6  {
7      public static void main(String[] args)
8      {
9          greetings("Donald");
10     }
11
12
13    /**
14     *       The greetings method displays a message
15     */
16     public static void greetings(String name)
17     {
18         System.out.println("Hello " + name + " from a method!");
19     }
20 }
```

Your output should be similar to as shown below:

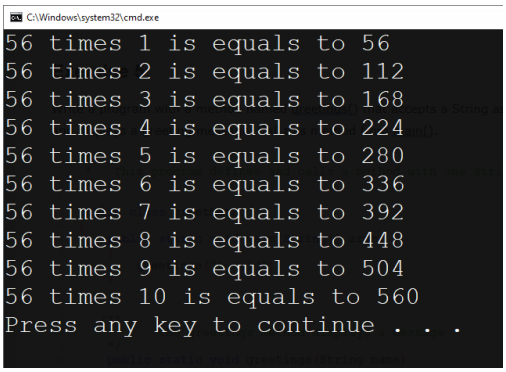


Exercise 6

Write a program with a void method named **printTables()** that accepts an integer as an argument and displays the times tables for the value provided, eg:

```
printTable(56);
```

will produce the following output:

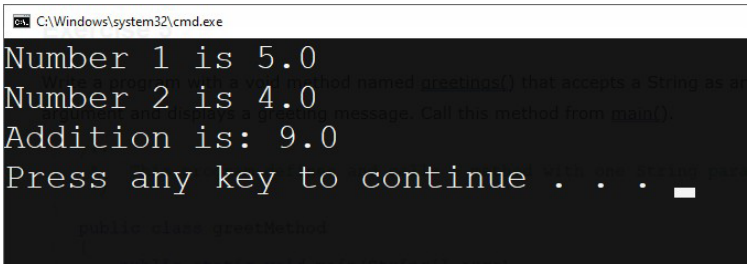


Exercise 7

Write a program with a void method named **addNumbers()** that accepts two doubles as arguments. The method should add the two doubles and output the result to the console.

```
addNumbers(5, 4);
```

Will produce the following output:

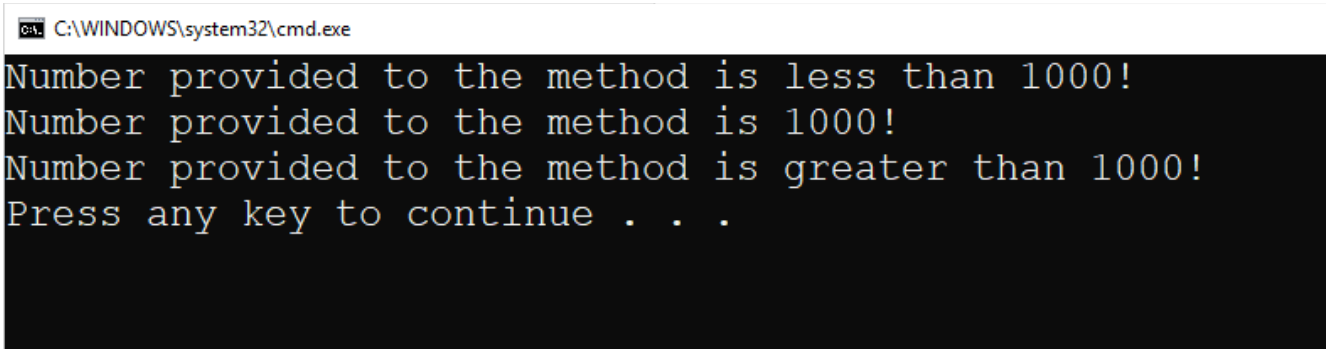


Exercise 8

Write a program with a void method named **checkNum()** that accepts a single integer as an argument. The method should output a message stating whether or not the number is greater than or less than 1000. If the number is exactly 1000, the an appropriate message should be output stating this. For example, if your main method is as below:

```
1  public class exercise8
2  {
3      public static void main(String[] args)
4      {
5
6          checkNum(95);
7
8          checkNum(1000);
9
10         checkNum(4567);
11
12     }
```

The output should be as follows:

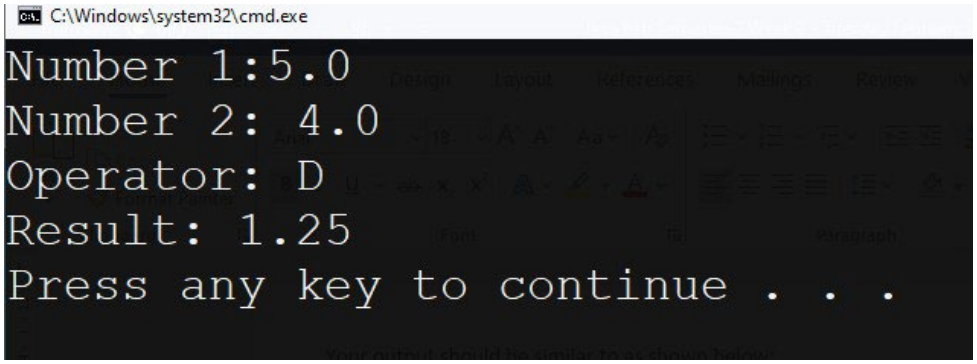


Exercise 9

Write a program with a void method named **calculate()** that accepts two doubles and a character as arguments. The two doubles denote numbers that can be provided to perform calculations on, and the character specifies the operator, where A is for addition, S is for subtraction, D is for division, and M is for multiplication (uppercase or lowercase characters should be allowed). If an incorrect operator is specified, then message should be output to the console, stating "This is not a valid operator!".

```
calculate(5, 4, 'D');
```

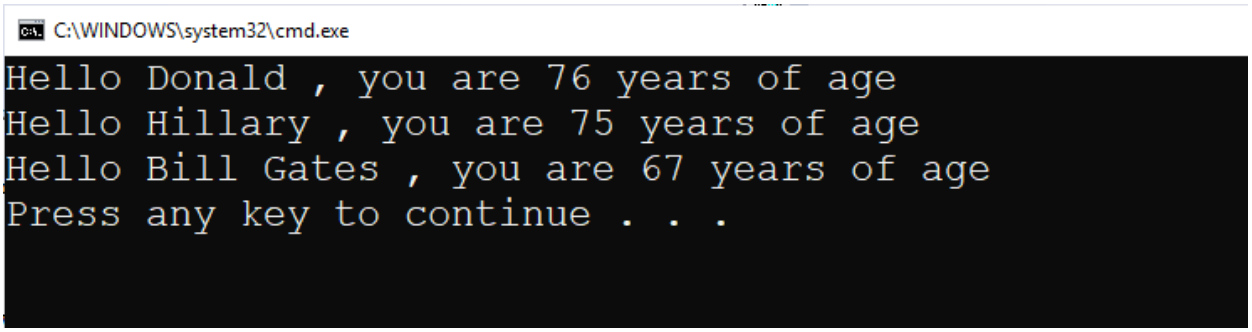
Running the code above in the main method should produce the following result:



Exercise 10

Write a program with a void method that accepts two arguments – one for a persons name and one for the year they were born. The method should benamed age(). Your main method should be similar to as shown below – write the associated method so that the output is as shown:

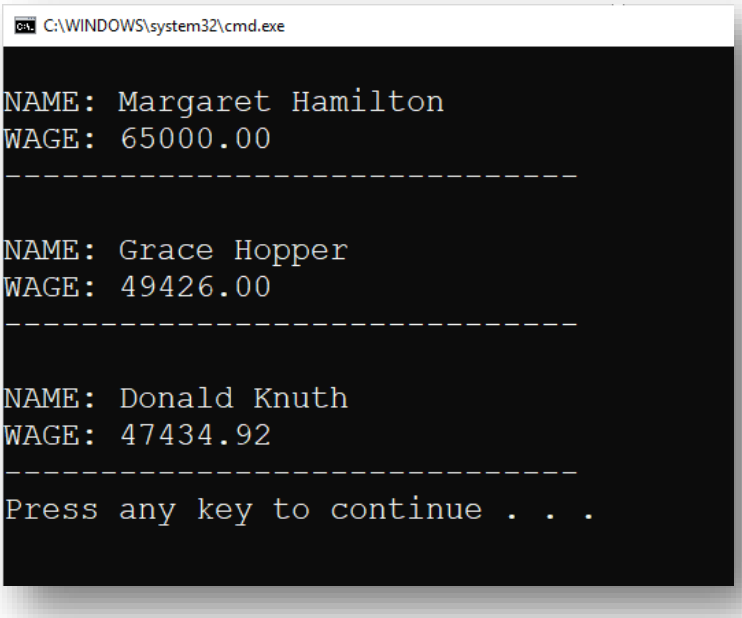
```
public class exercise10
{
    public static void main(String[] args)
    {

        age("Donald", 1946);

        age("Hillary", 1947);

        age("Bill Gates", 1955);

    }
```



Exercise 11

Write a program with a void method that accepts three arguments – one for a persons surname, one for the first name, and one for their weekly wage. The method should benamed calcAnnual(). The method should calculate the annual wage based on the weekly wage provided. Your main method should be similar to as shown below – write the associated method so that the output is as shown:

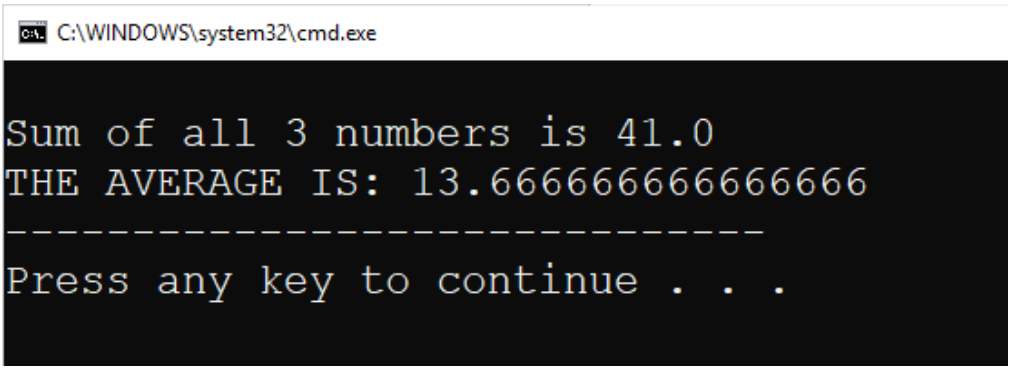
```
1  public class exercise11
2  {
3      public static void main(String[] args)
4      {
5
6          calcAnnual("Hamilton", "Margaret", 1250.00);
7
8          calcAnnual("Hopper", "Grace", 950.50);
9
10         calcAnnual("Knuth", "Donald", 912.21);
11
12     }
13
14
```



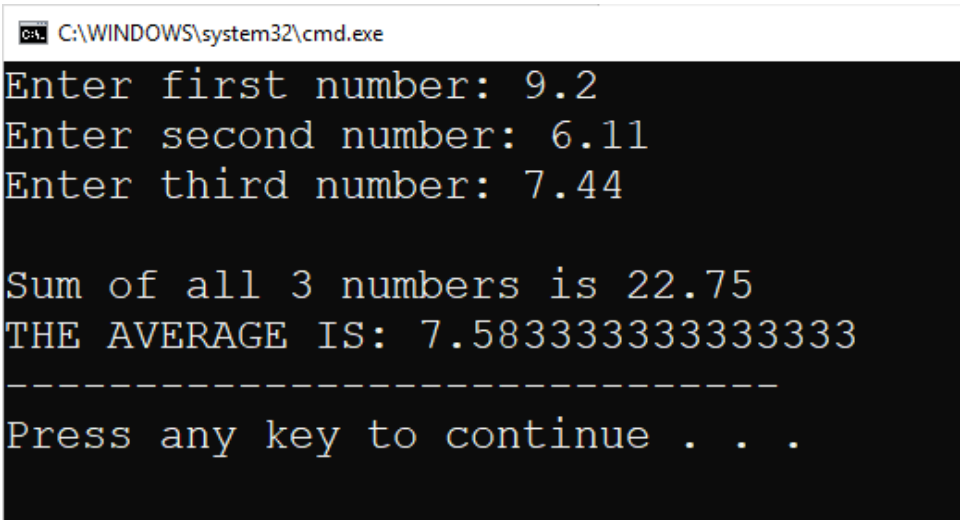
Exercise 12

Write a program with a void method that outputs the sum and average of 3 numbers (doubles). The method should benamed average() and it should accept three double as parameters. Your code in the main method should be similar to as shown below:

```
1  public class exercise12
2  {
3      public static void main(String[] args)
4      {
5
6          average(12, 14, 15);
7
8      }
9
```



Amend your code so that the user will be prompted to enter the three numbers via the console. You can do this via the main method – eg, using Scanner in the main method.



Exercise 13

Write a program with a method ***that returns an integer value***. The method should be named calc() and it should accept two integers as parameters. Your code should be similar to as shown below:

```
1  public class addReturn
2  {
3      public static void main(String[] args)
4      {
5
6          int myVar = calc(19, 2);
7
8          System.out.println(myVar);
9
10     }
11
12     public static int calc(int num1, int num2)
13     {
14         int result;
15         result = num1 + num2;
16
17         return result;
18     }
19
20 }
```

Exercise 14

Write a method named timesTen(). The method should accept a double argument, and return a doublevalue that is ten times the value of the argument.

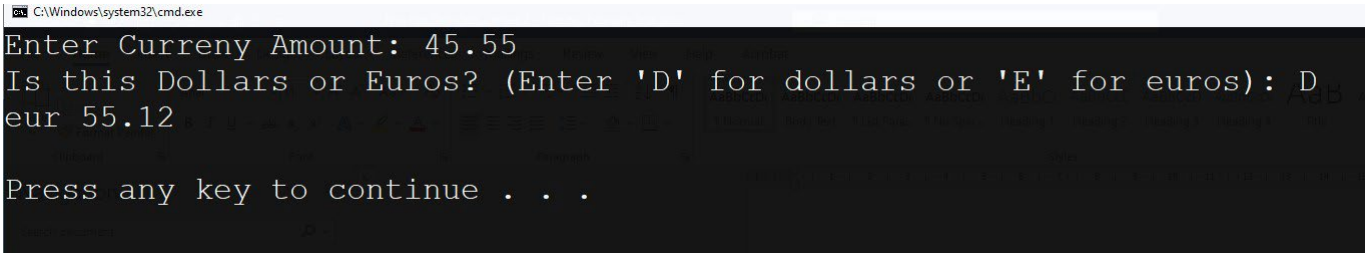
In the same program, write a method named square() that accepts an integer argument and returns thesquare of that argument.

In the same program, write a method named getName() that prompts the user to enter his or her first name,and then returns the user's input.

Exercise 15

Write a method named eurosToDollars(). The method should accept a double argument that is a number ofeuros, and return the equivalent number of dollars as a double. Assume that €1 is equals to $1.21.

Add a second method in the same program called dollarsToEuros(), that will convert accordingly.



Exercise 16

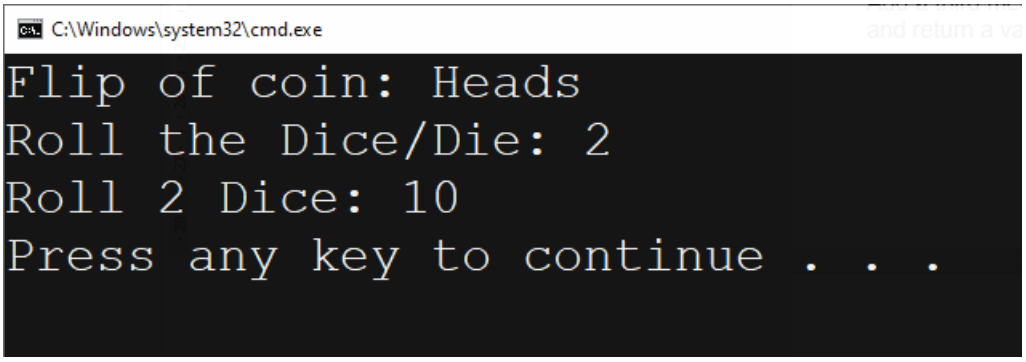
Write a program with a method that **returns a double** value. The method should be named calculate() and accept two doubles and a character as arguments. The two doubles denote numbers that can be provided to perform calculations on, and the character specifies the operator, where A is for addition, S is for subtraction, D is for division, and M is for multiplication (uppercase or lowercase characters should be allowed). The result should be returned as a double.

Exercise 17

Write a program with a method that returns a String value called coinflip(). The method should simulatetossing a coin and return either the string value “heads” or “tails”.

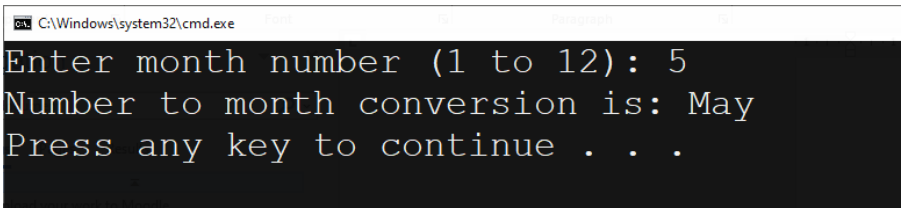
Add an addition method in the same program called rollDice() that returns an int value. The method shouldsimulate rolling a dice and return 1, 2, 3, 4, 5, or 6.

Add a third method called roll2Dice() that returns an int value. The method should simulate rolling 2 diceand return a value from 1 to 12. (You can use the rollDice() method to generate this value)

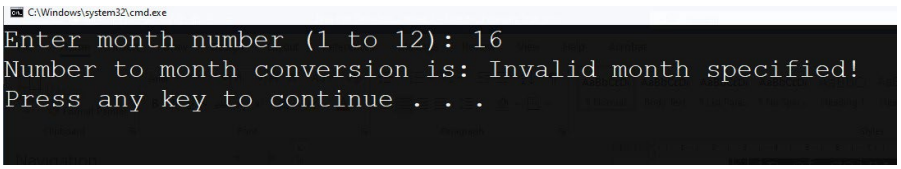


Exercise 18

Write a program with a method that will return the name of the appropriate month as a String when it has been given a number between one and 12 if a number lower or higher than 12 is provided , it should returnthe value “Invalid month specified!”.



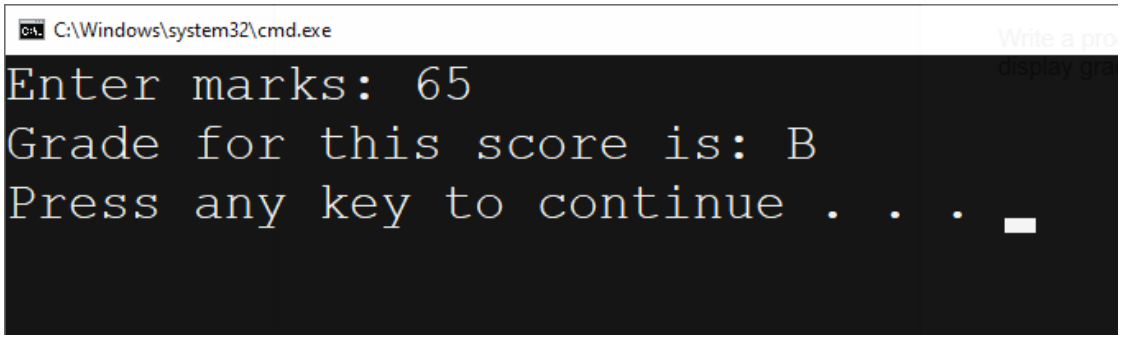
or

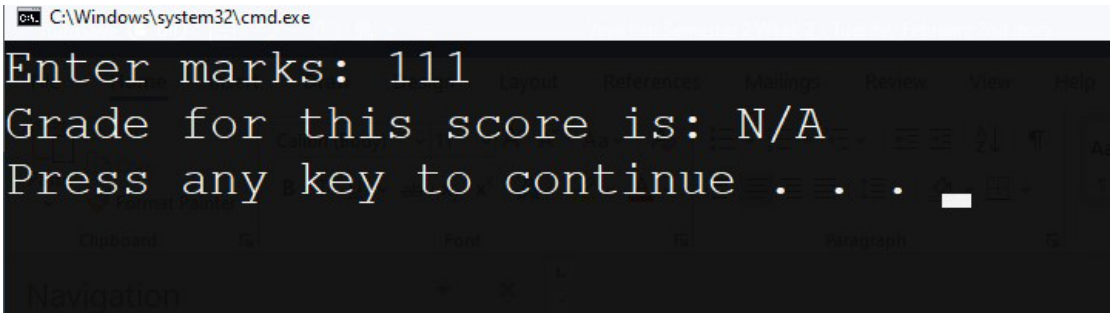


Exercise 19

Write a program which will prompt the user to enter his/her marks (out of 100). Define a method that willdisplay grades according to the marks entered as below:

Marks	Grade
85 - 100	A+
70-84	A
60-69	B
50-60	C
40-50	D
30-40	E





Exercise 20

Write a program that calculates the bill for a mobile telephone company. The company has three phoneplans, standard and premium. The rates are as follows:

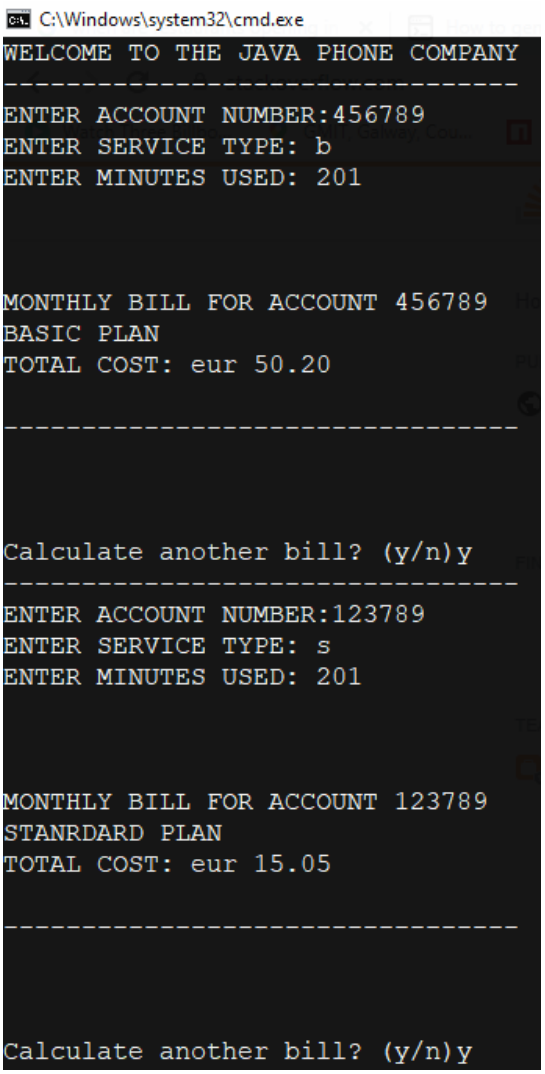
Basic:
€7.99 per month
All calls are charged at 0.21 cents per minute

Standard:
€10 per month
The first 100 minutes are free
All minutes thereafter are charged at 0.05 cents per minute

Premium:
€25 per month
The first 200 minutes are free
All minutes thereafter are charged at 0.01 cent per minute

Your program should allow staff at the company to calculate the monthly bill per customer. Your program should prompt for an account number, and service type (as a character – 'b' for basic, 's' for standard and'p' for premium), and the number of minutes used in the current month. Using this information, the bill should be presented to the user.

Your program should have separate methods to calculate the bill for each plan, and should return the cost as a double. The minutes used should be passed as an argument to the method as required.



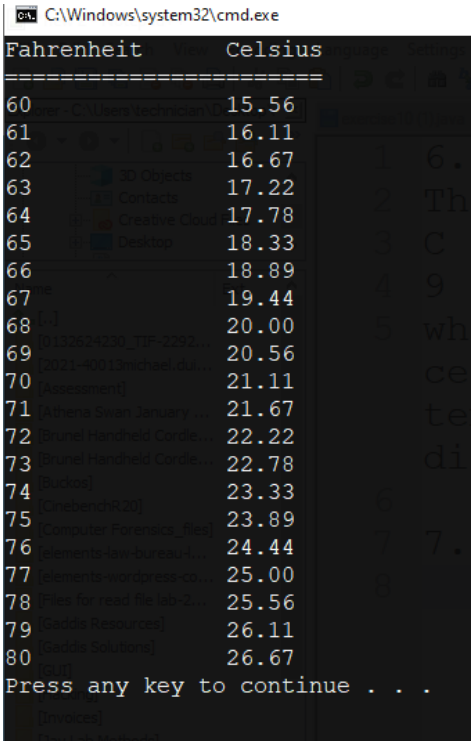
Exercise 21

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 that displays a table of the Fahrenheit temperatures 0 through 20 and their Celsius equivalents.

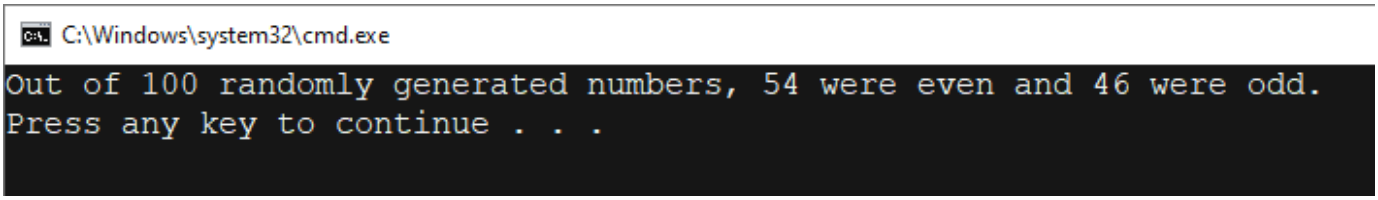


Exercise 22

You can use the following logic to determine whether a number is even or odd:

```
if ((number % 2) == 0)
{
// The number is even.
}
else
{
// The number is odd.
}
```

Write a program with a method named isEven() that accepts an *int* argument. The method should return the boolean true if the argument is even, or false otherwise. The program's main method should use a loop to generate 100 random integers. It should use the isEven method to determine whether each random number is even, or odd. When the loop is finished, the program should display the number of even numbers that were generated, and the number of odd numbers. Your output should be similar to as shown:

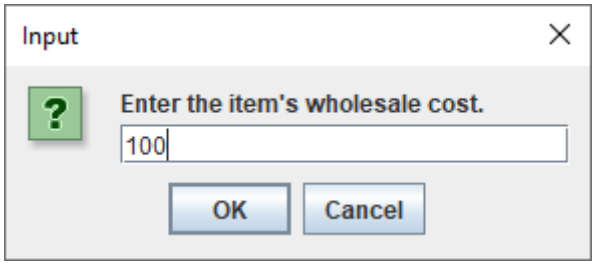


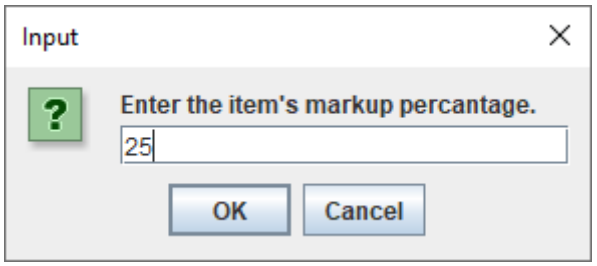
Exercise 23

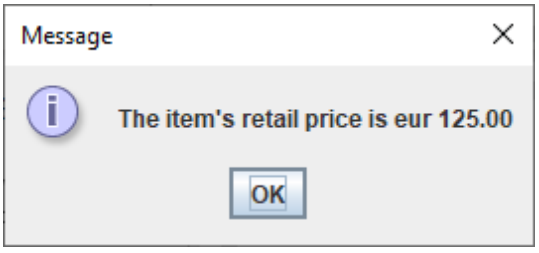
Write a program that asks the user to enter an item's wholesale cost and its markup percentage. It should then display the item's retail price. For example:

- 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:







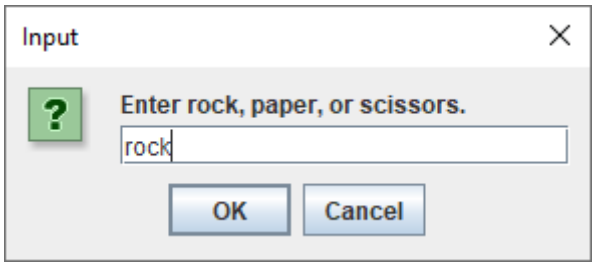
Exercise 24

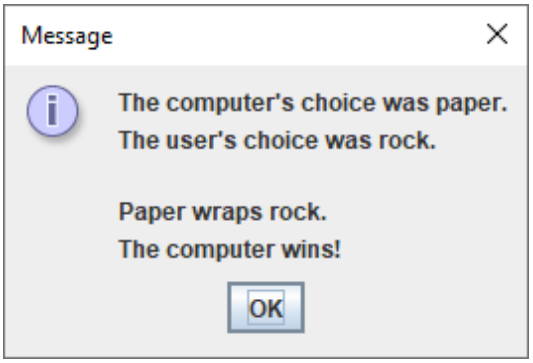
Write a program that lets the user play the game of Rock, Paper, Scissors against the computer. The program should work as follows.

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 a menu 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 rock smashes 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 wraps rock.)
    - If both players make the same choice, the game must be played again to determine the winner.

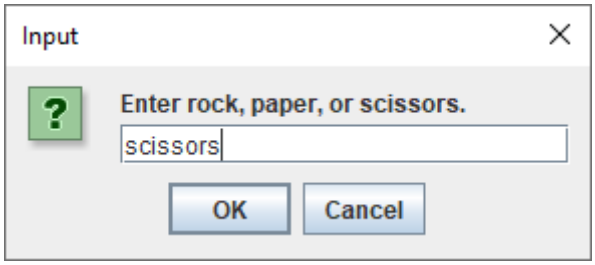
Be sure to divide the program into methods that perform each major task.

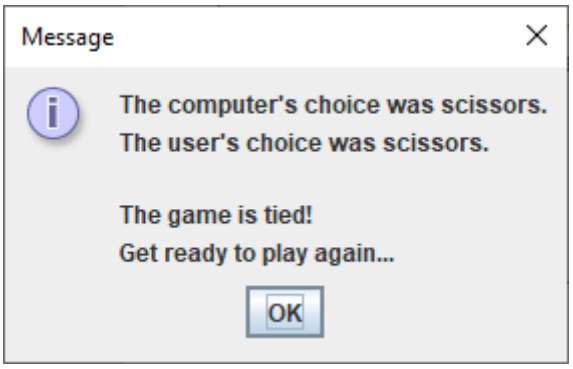
Your output should be similar to as shown below:



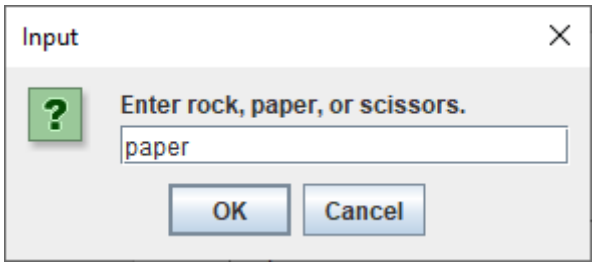


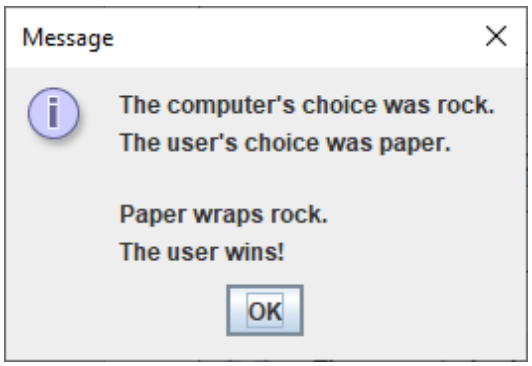
or


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