



## 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.

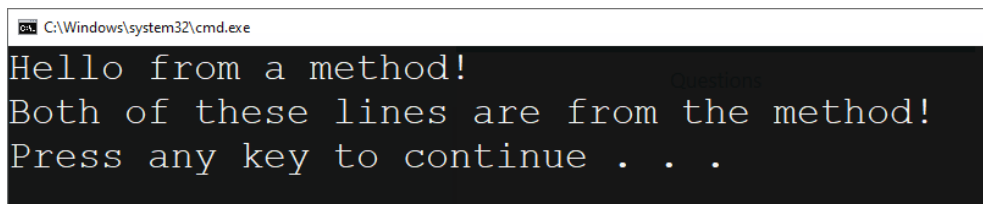
The code below shows how to create a simple *void* method that outputs a message. Note that the method is from lines 9 to 12, and the method is called from inside the main method, on line 6.

```
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:



C:\Windows\system32\cmd.exe

```
Hello from a method!
Both of these lines are from the method!
Press any key to continue . . .
```

## Exercise 3

```
voidMethods.java * X
1  public class voidMethods
2  {
3      public static void main(String[] args)
4      {
5          oscar();
6          coco();
7          john();
8          francis();
9      }
10
11 }
```

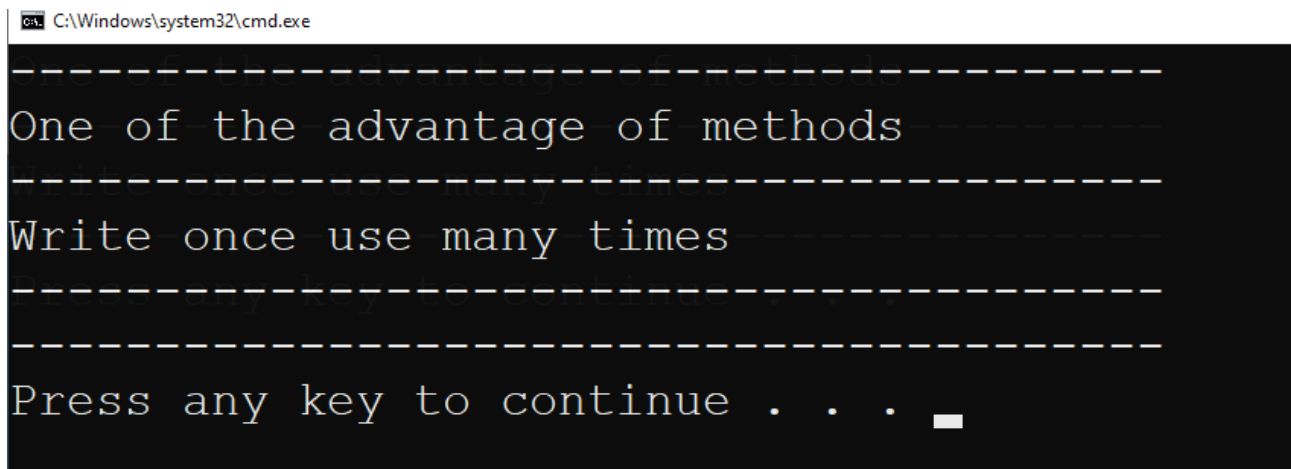
C:\Windows\system32\cmd.exe

```
Experience is the name everyone gives to their mistakes. - Oscar Wilde
In order to be irreplaceable, one must always be different. - Coco Chanel
First, solve the problem. Then, write the code. - John Johnson
Knowledge is power. - Francis Bacon
Press any key to continue . . .
```

## 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:



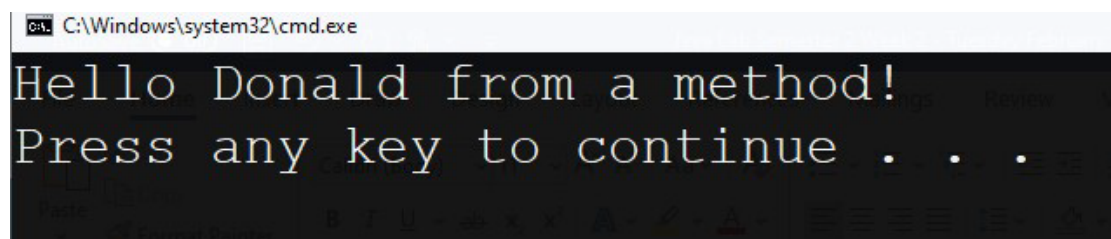
```
C:\Windows\system32\cmd.exe
-----
One of the advantage of methods
-----
Write once use many times
-----
-----
Press any key to continue . . . _
```

## 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:



The screenshot shows a Windows command prompt window with the title bar "C:\Windows\system32\cmd.exe". The window contains the following text:

```
Hello Donald from a method!
Press any key to continue . . .
```

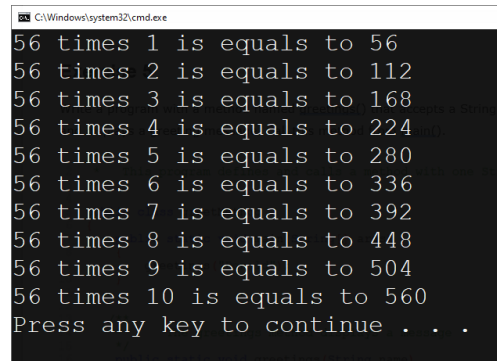
Below the text, there are several keyboard shortcuts and icons, including a mouse cursor icon, a keyboard icon, and a "Format Painter" icon.

## 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:*



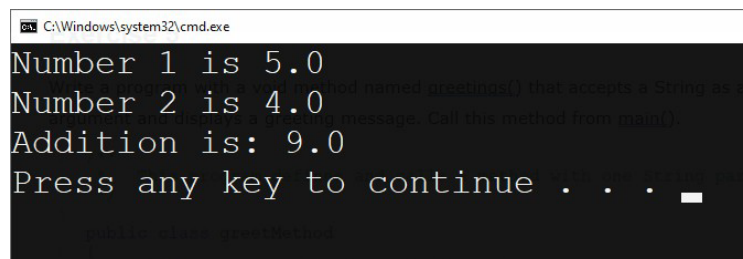
```
C:\Windows\system32\cmd.exe
56 times 1 is equals to 56
56 times 2 is equals to 112
56 times 3 is equals to 168
56 times 4 is equals to 224
56 times 5 is equals to 280
56 times 6 is equals to 336
56 times 7 is equals to 392
56 times 8 is equals to 448
56 times 9 is equals to 504
56 times 10 is equals to 560
Press any key to continue . . .
```

## 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:*



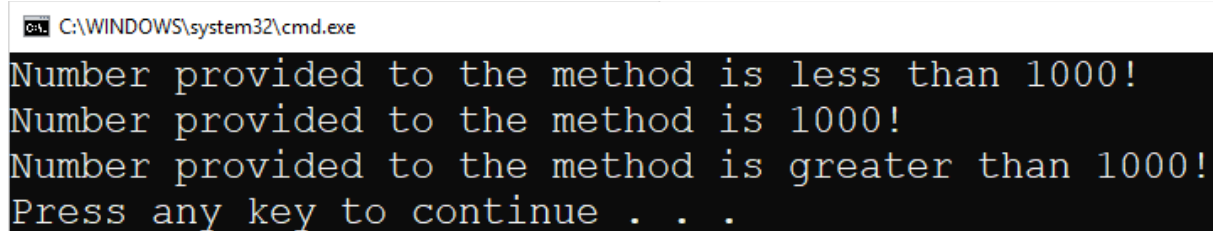
```
C:\Windows\system32\cmd.exe
Number 1 is 5.0
Number 2 is 4.0
Addition is: 9.0
Press any key to continue . . .
```

## 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    }
13 }
```

The output should be as follows:



C:\WINDOWS\system32\cmd.exe

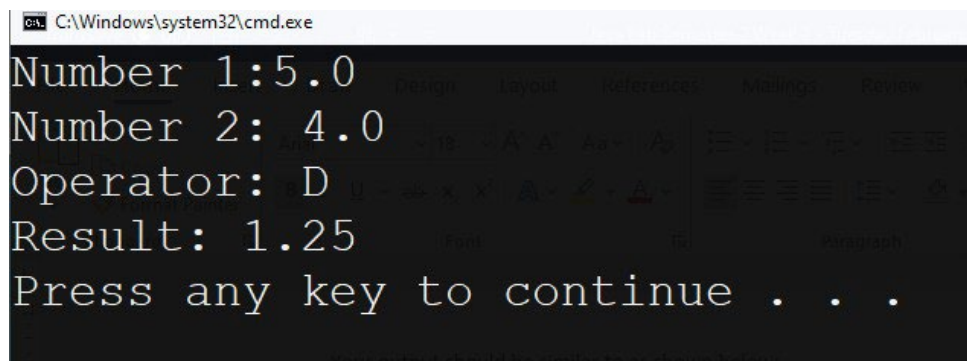
```
Number provided to the method is less than 1000!
Number provided to the method is 1000!
Number provided to the method is greater than 1000!
Press any key to continue . . .
```

## 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:



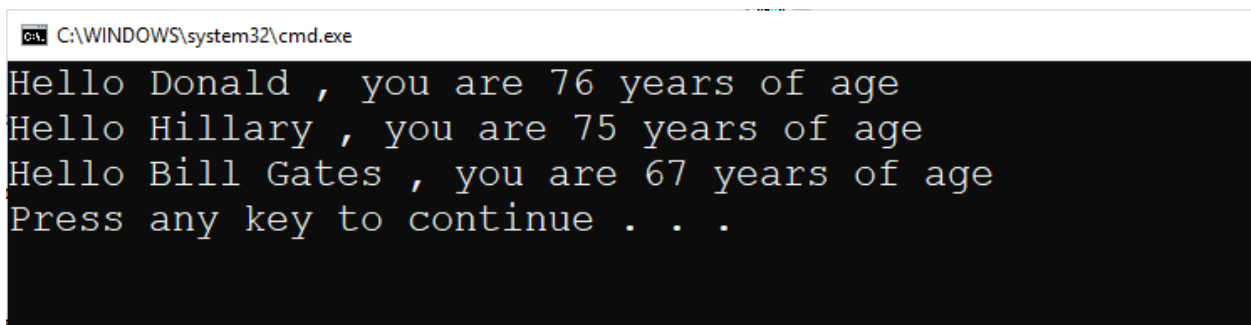
```
C:\Windows\system32\cmd.exe
Number 1:5.0
Number 2: 4.0
Operator: D
Result: 1.25
Press any key to continue . . .
```



## 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 be named age(). Your main method should be similar to as shown below – write the associated method so that the output is as shown:

```
public class exercisel0
{
    public static void main(String[] args)
    {
        age("Donald", 1946);
        age("Hillary", 1947);
        age("Bill Gates", 1955);
    }
}
```



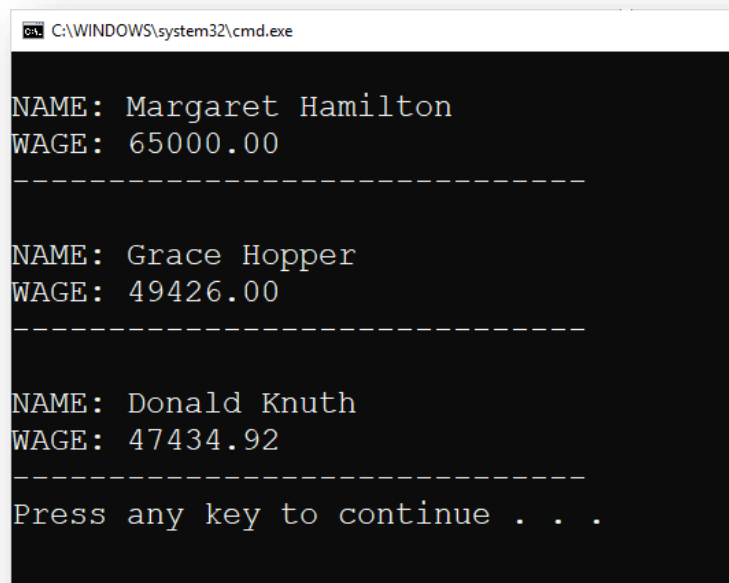
The screenshot shows a Windows command prompt window with the title bar "C:\WINDOWS\system32\cmd.exe". The window contains the following text:

```
Hello Donald , you are 76 years of age
Hello Hillary , you are 75 years of age
Hello Bill Gates , you are 67 years of age
Press any key to continue . . .
```

## 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 be named `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 exercisel1
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
```



```
C:\WINDOWS\system32\cmd.exe

NAME: Margaret Hamilton
WAGE: 65000.00
-----

NAME: Grace Hopper
WAGE: 49426.00
-----

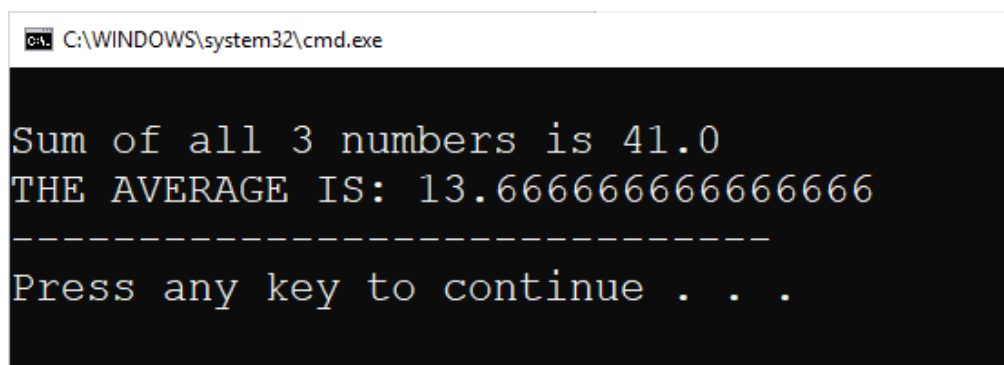
NAME: Donald Knuth
WAGE: 47434.92
-----

Press any key to continue . . .
```

## Exercise 12

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

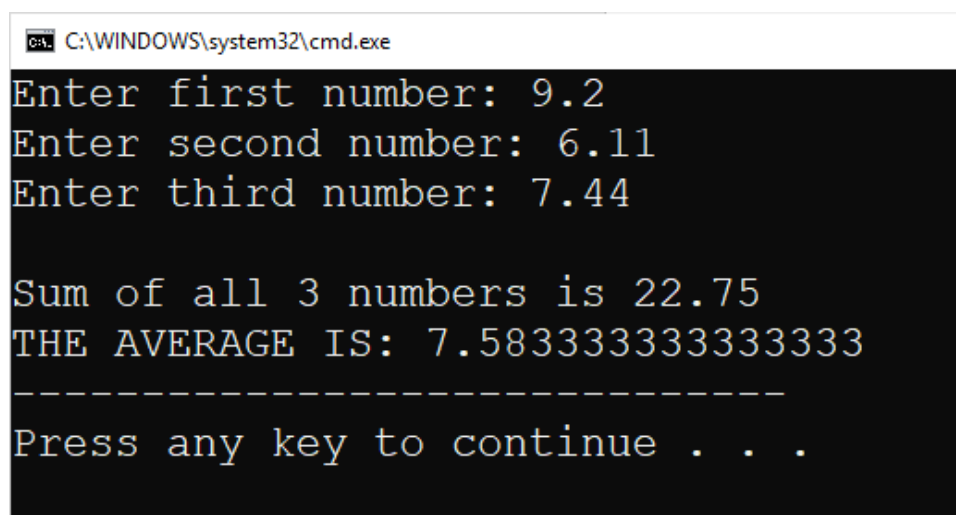
```
1 public class exercisel2
2 {
3     public static void main(String[] args)
4     {
5
6         average(12, 14, 15);
7
8     }
9
```



C:\WINDOWS\system32\cmd.exe

```
Sum of all 3 numbers is 41.0
THE AVERAGE IS: 13.666666666666666
-----
Press any key to continue . . .
```

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.



C:\WINDOWS\system32\cmd.exe

```
Enter first number: 9.2
Enter second number: 6.11
Enter third number: 7.44

Sum of all 3 numbers is 22.75
THE AVERAGE IS: 7.583333333333333
-----
Press any key to continue . . .
```

## 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 double value that is ten times the value of the argument.

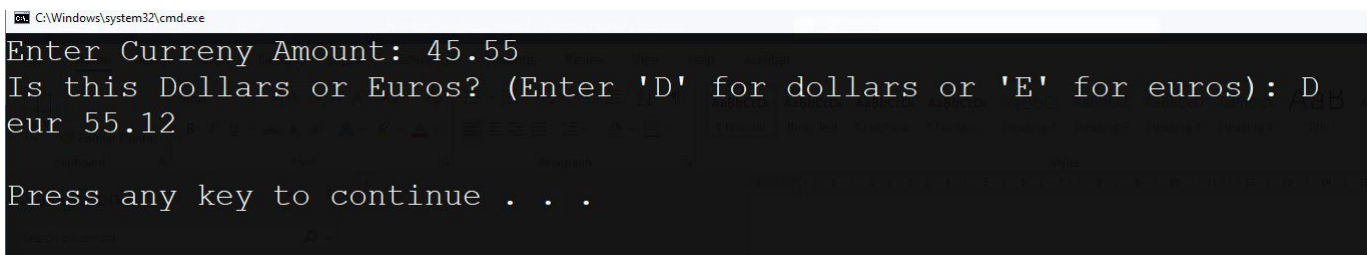
In the same program, write a method named `square()` that accepts an integer argument and returns the square of that argument.

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 of euros, and return the equivalent number of dollars as a double. Assume that €1 is equal to \$1.21.

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



```
C:\Windows\system32\cmd.exe
Enter Currency Amount: 45.55
Is this Dollars or Euros? (Enter 'D' for dollars or 'E' for euros): D
eur 55.12

Press any key to continue . . .
```

## Exercise 16

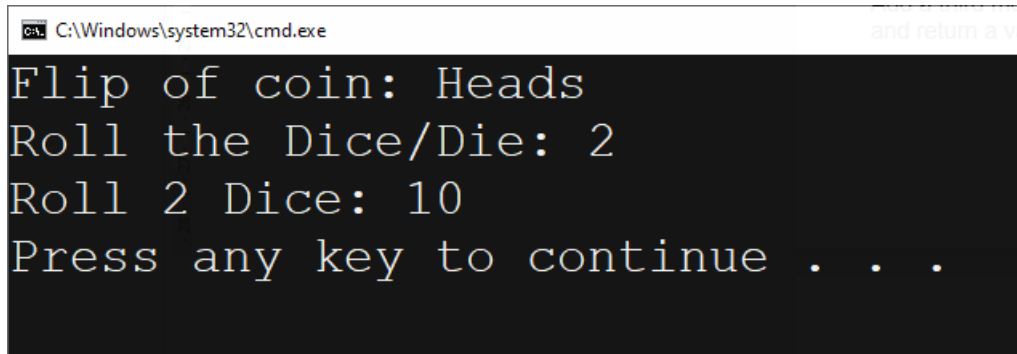
Write a program with a method that **returns a double** value. The method should be named `calculate()` and accept two doubles and a character as arguments. The two doubles denote numbers that can be provided to perform calculations on, and the character specifies the operator, where A is for addition, 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 simulate tossing 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 should simulate rolling a dice and return 1, 2, 3, 4, 5, or 6.

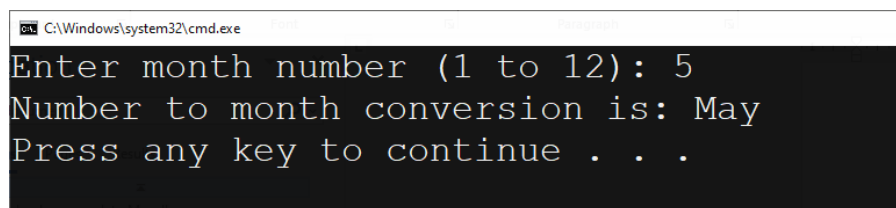
Add a third method called `roll2Dice()` that returns an int value. The method should simulate rolling 2 dice and return a value from 1 to 12. (You can use the `rollDice()` method to generate this value)



```
C:\Windows\system32\cmd.exe
Flip of coin: Heads
Roll the Dice/Die: 2
Roll 2 Dice: 10
Press any key to continue . . .
```

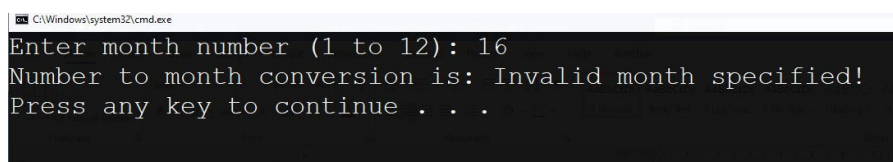
## Exercise 18

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 return the value "Invalid month specified!".



```
C:\Windows\system32\cmd.exe
Enter month number (1 to 12): 5
Number to month conversion is: May
Press any key to continue . . .
```

or

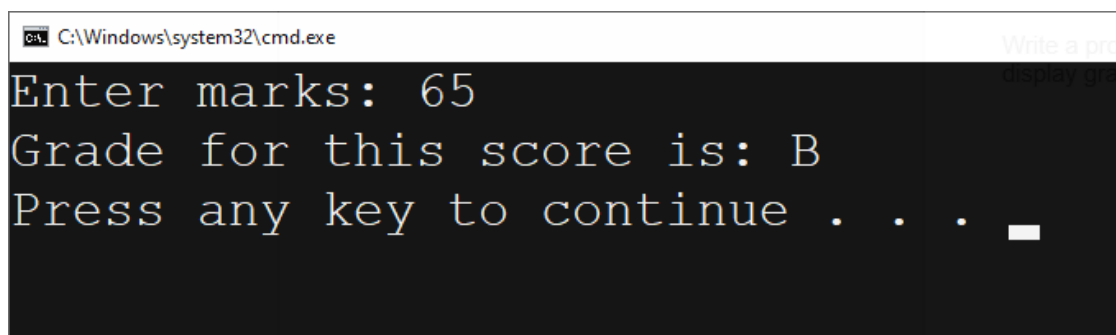


```
C:\Windows\system32\cmd.exe
Enter month number (1 to 12): 16
Number to month conversion is: Invalid month specified!
Press any key to continue . . .
```

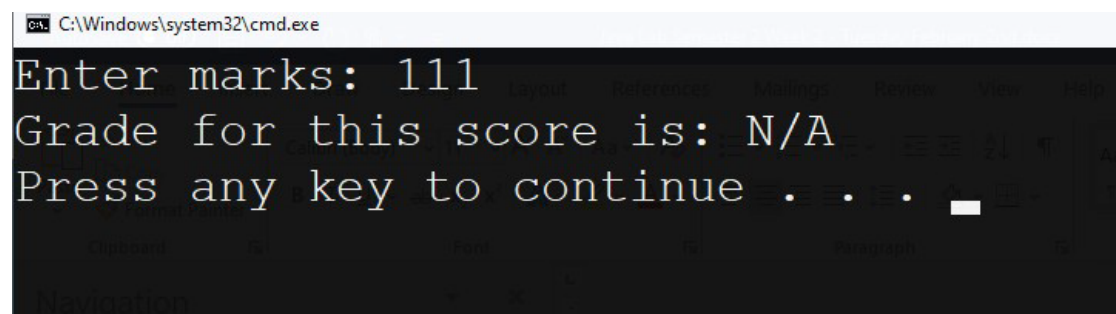
## Exercise 19

Write a program which will prompt the user to enter his/her marks (out of 100). Define a method that will display 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



```
C:\Windows\system32\cmd.exe
Enter marks: 65
Grade for this score is: B
Press any key to continue . . .
```



```
C:\Windows\system32\cmd.exe
Enter marks: 111
Grade for this score is: N/A
Press any key to continue . . .
```

## 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.

```
C:\Windows\system32\cmd.exe
WELCOME TO THE JAVA PHONE COMPANY
-----
ENTER ACCOUNT NUMBER:456789
ENTER SERVICE TYPE: b
ENTER MINUTES USED: 201

MONTHLY BILL FOR ACCOUNT 456789
BASIC PLAN
TOTAL COST: eur 50.20
-----

Calculate another bill? (y/n)y
-----
ENTER ACCOUNT NUMBER:123789
ENTER SERVICE TYPE: s
ENTER MINUTES USED: 201

MONTHLY BILL FOR ACCOUNT 123789
STANRDARD PLAN
TOTAL COST: eur 15.05
-----

Calculate another bill? (y/n)y
```



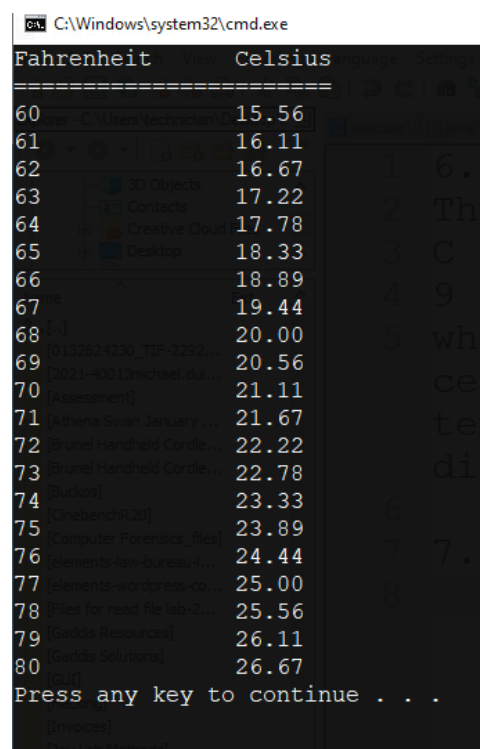
# Exercise 21

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

$$0.55555 \times (\text{tempInFahrenheit} - 32)$$

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.

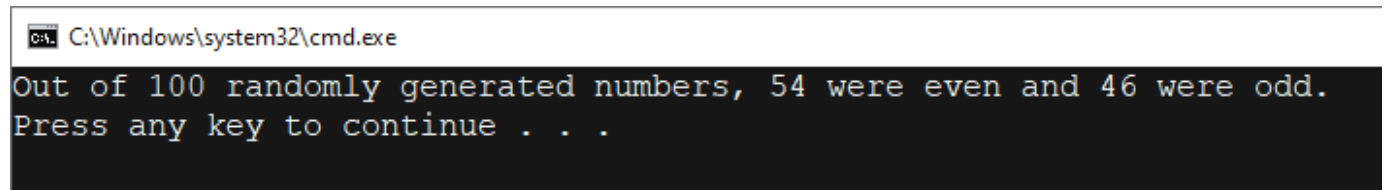


## Exercise 22

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:



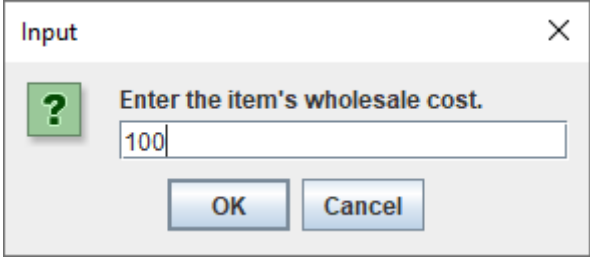
```
C:\Windows\system32\cmd.exe
Out of 100 randomly generated numbers, 54 were even and 46 were odd.
Press any key to continue . . .
```

## Exercise 23

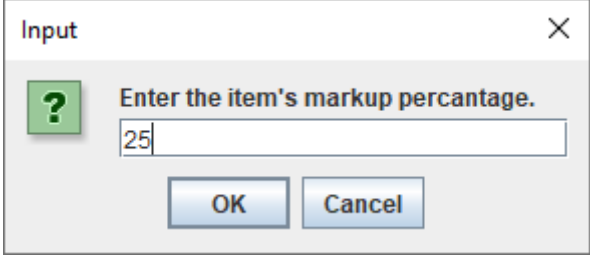
Write a program that asks the user to enter an item's wholesale cost and its markup percentage. It should then display the item's retail price. For example:

- 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.

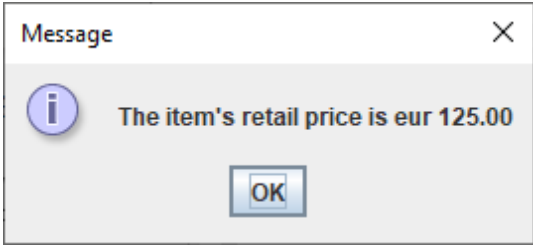
The program should have a method named `calculateRetail()` that receives the wholesale cost and the markup percentage as arguments, and returns the retail price of the item. Your output should be similar to as shown:



Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Enter the item's wholesale cost.", a text input field containing "100", and "OK" and "Cancel" buttons.



Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Enter the item's markup percentage.", a text input field containing "25", and "OK" and "Cancel" buttons.



Message dialog box titled "Message" with a close button (X). It contains a blue information icon (i), the text "The item's retail price is eur 125.00", and an "OK" button.

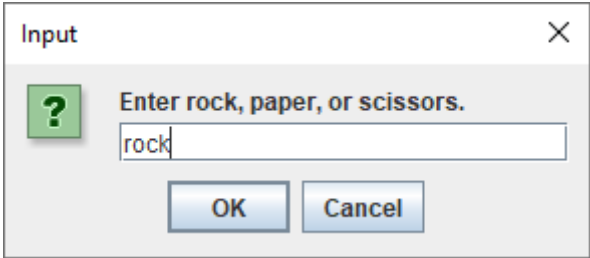
## Exercise 24

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

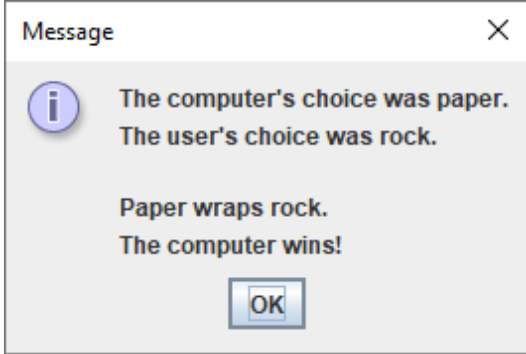
1. When the program begins, a random number in the range of 1 through 3 is generated. If the number is 1, then the computer has chosen rock. If the number is 2, then the computer has chosen paper. If the number is 3, then the computer has chosen scissors. (Don't display the computer's choice yet.)
2. 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:

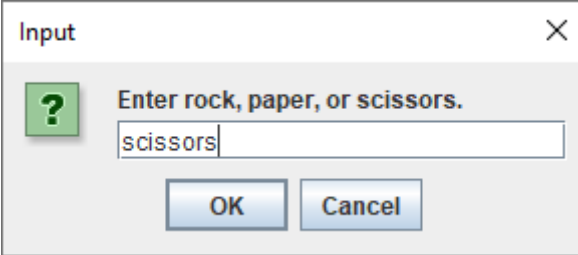


A screenshot of a Java-style input dialog box titled "Input". It has a close button (X) in the top right corner. On the left is a green square icon with a white question mark. To the right of the icon is the text "Enter rock, paper, or scissors." Below this text is a text input field containing the word "rock". At the bottom of the dialog are two buttons: "OK" and "Cancel".



A screenshot of a Java-style message dialog box titled "Message". It has a close button (X) in the top right corner. On the left is a blue circular icon with a white lowercase 'i'. To the right of the icon is the text "The computer's choice was paper." followed by "The user's choice was rock." Below this is the text "Paper wraps rock." followed by "The computer wins!". At the bottom of the dialog is a single button labeled "OK".

or



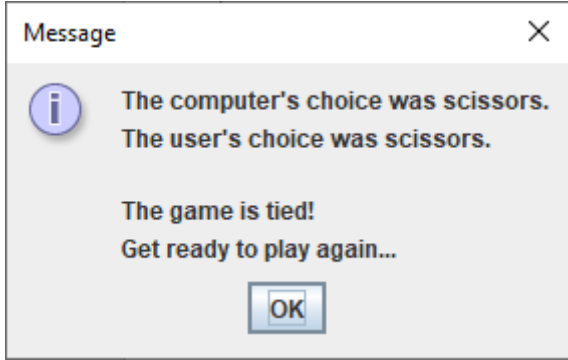
Input

Enter rock, paper, or scissors.

scissors

OK Cancel

This is a standard Windows-style input dialog box. It has a title bar with the word "Input" and a close button (X). The main area contains a green question mark icon, the prompt "Enter rock, paper, or scissors.", a text input field containing the word "scissors", and two buttons at the bottom: "OK" and "Cancel".



Message

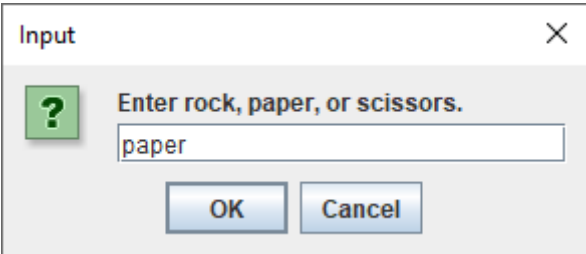
The computer's choice was scissors.  
The user's choice was scissors.

The game is tied!  
Get ready to play again...

OK

This is a standard Windows-style message dialog box. It has a title bar with the word "Message" and a close button (X). The main area contains a blue information icon (i), the text "The computer's choice was scissors. The user's choice was scissors.", the text "The game is tied! Get ready to play again...", and an "OK" button at the bottom.

or



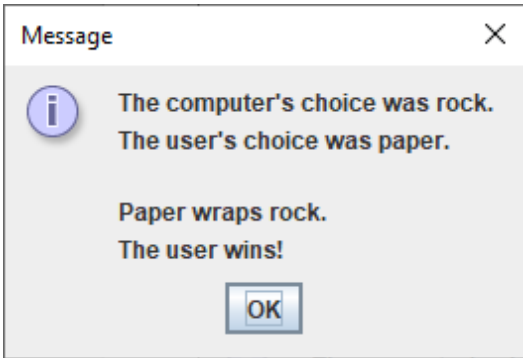
Input

Enter rock, paper, or scissors.

paper

OK Cancel

This is a standard Windows-style input dialog box. It has a title bar with the word "Input" and a close button (X). The main area contains a green question mark icon, the prompt "Enter rock, paper, or scissors.", a text input field containing the word "paper", and two buttons at the bottom: "OK" and "Cancel".



Message

The computer's choice was rock.  
The user's choice was paper.

Paper wraps rock.  
The user wins!

OK

This is a standard Windows-style message dialog box. It has a title bar with the word "Message" and a close button (X). The main area contains a blue information icon (i), the text "The computer's choice was rock. The user's choice was paper.", the text "Paper wraps rock. The user wins!", and an "OK" button at the bottom.