

Recap Variables, Methods, Conditional statements

Objectives

- Recap previous session
- Recap variables + exercises
- Recap methods + exercises
- More conditional statements exercises
- Homework exercises

Recap previous session

- What is the type of data that we can use to store `true` or `false` values?
- Can you give an example where this type really helps?
- What do we need in order to read from the Standard Input?
- How can we read `int`, `double` or `String` values?
- Enumerate the boolean operators that comes into your mind.
- What are the logical operators that we learned about?
- How do we make a check in JAVA? For example, if I want to execute a certain statement only if a condition is met, what should I use?
- How many conditional statements can I chain?
- What operator should I use if I want to subtract the remainder of a division?
- Can I have an If statement inside another If statement? What is this called?

Recap variables + exercises

- What is a variable?
- Why do we use variables?
- Can you tell the parts of a variable?
- Can you declare a variable without assigning it a value?
- What should you do before using a variable?

Exercises

1. Create A String variable and do not set a value to it. Now, print this variable. After you fix the issues, call the length method.
 - Solution:

```
public class Application {  
  
    public static void main(String[] args) {  
        String name = null;  
        System.out.println(name);  
        name.length();  
    }  
}
```

2. Declare a boolean variable which will hold the result between the comparison of 5 and 4 using the `smaller than` operator. Display the value of the variable using the `println` method.

◦ Solution:

```
public class Application {  
  
    public static void main(String[] args) {  
        boolean comparison = 5 < 4;  
        System.out.println(comparison);  
    }  
}
```

3. Declare a `double` variable and initialize it with a value of your choice (make sure it is a floating point variable.) Next declare a variable of type `int` and store the value of the `double` variable inside it. Next, print the value of the `int` variable.

◦ Solution:

```
public class Application {  
  
    public static void main(String[] args) {  
        double grade = 55.9;  
        int truncatedGrade = (int) grade;  
        System.out.println(truncatedGrade);  
    }  
}
```

- Recap methods + exercises

- What is a method?
- What is the most important method that any JAVA application has it?
- What is the structure of a JAVA method? Give some examples, define random methods.
- Can you have methods that have the same name but different number of arguments?
- Can you have method that have the same name, same number of arguments but different return type?

Exercises

1. Create a JAVA method which displays a message of your choice.

◦ Solution:

```
public class Application {  
  
    public static void display() {  
        System.out.println("Simple message");  
    }  
}
```

2. Create a JAVA method which displays a message which is received as a parameter of type `String`

- Solution

```
public class Application {  
  
    public static void display(String message) {  
        System.out.println(message);  
    }  
}
```

3. Create a JAVA method, which receives 3 double parameters and returns the smallest one. Do not use `if` statements (Hint: Use `Math.min` method)

- Solution

```
public class Application {  
  
    public static double min(double x, double y, double z) {  
        return Math.min(Math.min(x, y), z);  
    }  
}
```

4. Create a JAVA method which receives a `String` and an integer number as parameters. The method should return a new String where the character at the index represented by the integer parameter, has been removed. Note: the index should be in the interval 0, string length -1, inclusive.

- Sample Input: "Kitten", 1
- Sample Output: "Ktten"
- Sample Input2: "Kitten", 0
- Sample Output2: "itten"
- Solution:

```
public class Application {  
  
    public static void main(String[] args) {  
        System.out.println(removeAtPosition("Kitten", 1));  
        System.out.println(removeAtPosition("Kitten", 0));  
    }  
  
    public static String removeAtPosition(String text, int position) {  
        String front = text.substring(0, position);  
        String back = text.substring(position+1);  
        return front + back;  
    }  
}
```

More conditional statements exercises

1. Write a JAVA program which reads a letter from the Standard Input. The program should print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.

◦ Solution:

```
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a letter");
        String letter = in.next();
        if(letter.length() > 1 || Character.isDigit(letter.charAt(0)))
        {
            System.out.println("You should enter a letter!!");
        } else if(letter.equalsIgnoreCase("a") ||
letter.equalsIgnoreCase("e") || letter.equalsIgnoreCase("i") ||
letter.equalsIgnoreCase("o") || letter.equalsIgnoreCase("u")) {
            System.out.println("Vowel!");
        } else {
            System.out.println("Consonant");
        }
    }
}
```

2. Write a JAVA program which allows the user to enter a number between 1 and 10. Next, generate a random number between 0 and 10 and compare it against the number introduced by the user. If the number entered by user is greater, print "Too high", else if the number is smaller, print "Too small" and if the numbers are equal, press "You hit the JackPot". Note, in order to generate a random number between 1 and 10 use the following JAVA snippet:

```
Random random = new Random();
int randomNumber = random.nextInt(11);
```

Bonus: Extra POINTS if you create a method that validates the input entered by the user, meaning that the user is only allowed to enter a number between 0 and 10 inclusive

◦ Solution:

```

import java.util.Random;
import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a number between 0 and 10: ");
        int number = in.nextInt();

        if(isInputValid(number)) {
            Random random = new Random();
            int randomNumber = random.nextInt(11);

            if(number > randomNumber) {
                System.out.println("Too High!");
            } else if (number < randomNumber) {
                System.out.println("Too Low!");
            } else {
                System.out.println("You hit the jackpot!!");
            }

        } else {
            System.out.println("Bad Input! Only allowed to enter
numbers between 0 and 10");
        }
    }

    public static boolean isInputValid(int number) {
        return number >=0 && number <=10;
    }
}

```

3. Write a JAVA application which receives two `int` values from the user and asserts whether the first number is a multiple of the second one in the following format:

- `First number is a multiple of the second number` (if the first number is a multiple of the second one)
- `First number is not a multiple of the second number` (if the first number is not a multiple of the second one).

Also, the program should display `false` from the beginning if the second number happens to be 0.

- Solution:

```

import java.util.Scanner;

public class Application {

    public static void main(String[] args) {

```

```

Scanner in = new Scanner(System.in);
System.out.print("Enter the first number: ");
int firstNumber = in.nextInt();
System.out.print("Enter the second number: ");
int secondNumber = in.nextInt();

if(secondNumber == 0) {
    System.out.println("False!");
} else {
    if(firstNumber % secondNumber == 0) {
        System.out.println("First Number is a multiple of
the second number!");
    } else {
        System.out.println("First Number is not a multiple
of the second number!");
    }
}

}

public static boolean isInputValid(int number) {
    return number >=0 && number <=10;
}
}

```

4. Create a JAVA program which receives three double values, meaning the possible sides of a right triangle, and return wheter or not the three values can form a right-angled triangle.

◦ Solution:

```

import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter side a length: ");
        double sideA = in.nextDouble();
        System.out.print("Enter side b length: ");
        double sideB = in.nextDouble();
        System.out.print("Enter side c length: ");
        double sideC = in.nextDouble();

        //a b and c should be greater than 0

        boolean validRightTriangle = canFormRightTriangle(sideA, sideB,
sideC);
        System.out.println(validRightTriangle);
    }

    public static boolean canFormRightTriangle(double sideA, double

```

```

    sideB, double sideC) {
        if(sideA <= 0 || sideB <= 0 || sideC <= 0) {
            return false;
        }

        if( (sideA + sideB) <= sideC || (sideA + sideC) <= sideB ||
(sideB + sideC) <= sideA) {
            return false;
        }

        boolean result = (sideA * sideA + sideB * sideB) == sideC *
sideC;

        return result;
    }
}

```

5. Monkey problem. We have two monkeys, `a` and `b`, and two parameters `aSmile` and `bSmile` which indicates if each monkey is smiling. We are in trouble if both of them are smiling or if neither is smiling. Create a method which returns `true` if we are in trouble, `false` otherwise.

◦ Solution:

```

public class Application {

    public static void main(String[] args) {
        System.out.println(isMonkeyTrouble(true, true));
    }

    public static boolean isMonkeyTrouble(boolean aSmile, boolean
bSmile) {
        if(aSmile == true && bSmile == true || (aSmile == false &
bSmile == false)) {
            return true;
        } else {
            return false;
        }
    }
}

```

6. Try to reduce the logic for the previous exercises. Think in what scenarios do we return true and in what do we return false

◦ Solution:

```

public class Application {

    public static void main(String[] args) {
        System.out.println(isMonkeyTrouble(true, true));
    }
}

```

```

    }

    public static boolean isMonkeyTrouble(boolean aSmile, boolean
bSmile) {
        if(aSmile != bSmile) {
            return false;
        } else {
            return true;
        }
    }
}

```

7. We have a loud talking parrot. We are in trouble if our parrot is talking before hour 7 or after 20. Create a JAVA method which determines if we are in trouble or not. The method should receive two parameters:

- if the parrot is talking or not
- the hour (0-23)
- Solution:

```

public class Application {

    public static void main(String[] args) {
        System.out.println(isParrotTrouble(true, 19));
    }

    public static boolean isParrotTrouble(boolean isParrotTalking,
int hour) {
        if(isParrotTalking && (hour < 7 || hour > 20) ) {
            return true;
        } else {
            return false;
        }
    }
}

```

8. Create a JAVA method which receives a String and returns a new String where the first and the last characters have been exchanged.

- Sample Input: code
- Sample Output: eodc
- Sample Input: a
- Sample Output: a
- Sample Input: ab
- Sample Output: ba
- Solution:


```
public class Application {

    public static void main(String[] args) {
        System.out.println(reverseFrontAndBack("code"));
        System.out.println(reverseFrontAndBack("a"));
        System.out.println(reverseFrontAndBack("ab"));
    }

    public static String reverseFrontAndBack(String text) {
        if(text.length() <= 1) {
            return text;
        } else {
            String middle = text.substring(1, text.length()-1);
            return text.charAt(text.length()-1) + middle +
text.charAt(0);
        }
    }
}
```

Homework exercises

1. Create a JAVA method which receives a parameter of type String. The method should return a new String which will be the String received as a parameter + the word **not** added in front of it. Note: If the parameter already contains the word **not**, it should return the String unchanged.
 - Sample Input: "Candy"
 - Sample Output: "not Candy"
 - Sample Input 2: "not good"
 - Sample Output 2: "not good" Hint: Search for a method defined in the **String** class which verifies if a **String** starts with a certain text.
2. Create a JAVA method which receives a String as a parameter. The method will consider the first three characters of the text as the front of it and should return a new String which will contain three times the front of the parameter. Note: if the text contains less than 3 characters, the front is whatever is there.
 - Sample Input: JAVA
 - Sample Output: JAVJAVJAV
 - Sample Input2: Chocolate
 - Sample Output2: ChoChoCho
 - Sample Input3: Ab
 - Sample Output3: AbAbAb
3. Create a JAVA method which receives a String parameter and returns a new String with the last character added at the front and at the back. The text parameter should be at least of length 1.
 - Sample Input: cat
 - Sample Output: tcatt

- Sample Input2: Hello
 - Sample Output2: oHelloo
 - Sample Input3: a
 - Sample Output: aaa
4. Create a JAVA method which receives an integer parameters and returns true if the number is either a multiple of 3 or a multiple of 5.
- Sample Input: 3
 - Sample Output: true
 - Sample Input2: 10
 - Sample Input2: true
 - Sample Input3: 8
 - Sample Output3: false
5. Given two temperatures, return true if one is less than 0 and the other is greater than 100
- Sample Input: 120, -1
 - Sample Output: true
 - Sample Input2: -1, 120
 - Sample Output2: true
 - Sample Input3: 2, 120
 - Sample Output3: false
6. Given two int values, return true if either of them is in the range 10..20 inclusive
- Sample Input: 12, 99
 - Sample Output: true
 - Sample Input2: 21, 12
 - Sample Output2: true
 - Sample Input3: 8, 99
 - Sample Output3: false

Guidelines

- Try at the end of each session to note about each new JAVA class and methods that you encountered (e.g Random, Math, etc).
- Try to implement again all the exercises that have been implemented by us, together. You will see that you might come with a different solution! But this is not wrong! Because a problem, most of the time, can be solved in multiple ways.