

Questions and Exercises

These questions and exercises is an opportunity to see what you've learnt from the lecture as well as practice the new things we've been talking about. In other words, these questions and exercises are completely optional but it's recommended to do them. In the end of the document you will find the answers to the questions as well as possible solutions to the exercises, note that one can solve an exercise in different ways. There will also be some suggestions about what one could code if one want to continue with some more advanced things. These suggestions will not come with a possible solution and might include things that haven't been covered in the lecture.

Question 1

Which are the six relational operators? And how do one use them in code?

Question 2

What's wrong with the following code?

```
System.out.println(true && false || true !! true);
```

Question 3

Given the following code

```
int x = 5;
if (x > 10) {
    System.out.println("x is greater than 10");
}
```

how do you expand it to print another message if x is between 3 and 10 (inclusive).

Question 4

When you check if two Strings are equal to each other you'll have to do something special. What? How do you do it for two chars(characters)?

Question 5

What would the following code print out on the screen? Answer the question before you run the code.

```
int x = 20;
int y = 15;
System.out.println(x < y || y % 2 == 1 && x < 20);
```

Exercise 1

Let the user choose 3 cards by choosing from Ace(1) to King(13) by entering integers. If the user has either three of a kind or a pair you should print that out. The message should include what the user had three of a kind or pair of, use the numbers instead of the names. For instance if the user picks 1(Ace), 10 and 10 it should print out "You got a pair. (10)" or something similar.

Exercise 2

Write a program using if statements that asks the user for 4 integers and outputs the maximum value. For instance if the user enters 14, 3, -5 and 14 the program is supposed to output 14.

Exercise 3

Exclusive or, or XOR is a logical operator defined so it will return true only if one of the values is true. Given two boolean values p and q one can simply code it like the following

```
//will output p XOR q  
System.out.println(p != q);
```

Your job is however to write a program made from AND, OR and NOT that outputs p XOR q. The table below shows you what p XOR q is depending on p and q.

p	q	p XOR q
false	false	false
false	true	true
true	false	true
true	true	false

Further explorations

Using the following code will create a random number generator

```
import java.util.Random;  
Random myRandomGenerator = new Random();
```

When you have it you can get a random number between 0 and 9 (inclusive) like this

```
myRandomGenerator.nextInt(10);
```

Use this to write a program that randomizes one of the 5 arithmetical operators (+, -, *, / and %) and then gives the user a simple expression to solve, for instance "3 + 2". The user should then enter the integer solution and the program tells you if the answer was correct or not. In the case of the wrong answer the program should show the correct solution.

Make sure that you don't divide by 0.

Answers and solutions

Answer to Question 1

The following are the relational operators

Equals to ==

Not equals to !=

Greater than >

Greater than or equals to

Lesser than <

Lesser than or equals to <=

They can be used like this for instance

```
//Will output true since 5 is greater than 2
System.out.println(5 > 2);
```

Answer to Question 2

The NOT operator is first of all only using one exclamation point and second of all it's only operating on one value. There's no obvious way to fix the code in the question since it's tricky to know what the coder intended to do. However, the code below is an example on how one is supposed to use the NOT operator.

```
System.out.println(true && false || !true);
```

Answer to Question 3

Since we're already checking if x is greater than 10 we can simply add an else if statement to check if x is greater than or equals to 3. The code could look like the following.

```
int x = 5;
if (x > 10) {
    System.out.println("x is greater than 10");
}else if(x >= 3) {
    System.out.println("This is another message");
}
```

Now x has to be 10 or less for it to reach the x >= 3 if statement. This creates the upper limit of the 3 to 10 range while the x >= 3 creates the lower limit.

Answer to Question 4

When checking for equality of two strings one has to do something like the following

```
if (myStringVariable.equals("Dog")) {
    //do stuff
}
```

When checking for equality for two chars you just do it like you normally would

```
if (myCharVariable == 'V') {
    //do stuff
}
```

Answer to Question 5

The code from the question will output

false

Evaluating the relational operators will give you the following

```
System.out.println(false || true && false);
```

From the prioritizing rules you then have to begin with evaluating the AND operator

```
System.out.println(false || false);
```

And now it is easy since you get the following code

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

Possible solution to Exercise 1

```
import java.util.Scanner;
public class Exercise1 {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);

        System.out.println("Enter your three cards (1-13)");
        int card1 = myScanner.nextInt();
        int card2 = myScanner.nextInt();
        int card3 = myScanner.nextInt();

        //Check if the user has three of a kind
        if (card1 == card2 && card2 == card3) {
            System.out.println("You got three of a kind (" + card1 + ").");
        }
        //Check if the user has a pair where card1 is involved
        }else if(card1 == card2 || card1 == card3) {
            System.out.println("You got a pair (" + card1 + ").");
        }
        //Check if the user has a pair with last two cards.
        }else if(card2 == card3) {
            System.out.println("You got a pair (" + card2 + ").");
        }
    }
}
```

Possible solution to Exercise 2

```
import java.util.Scanner;
public class Exercise2 {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);

        //Store the 4 integers
        System.out.println("Please enter four integers");
        int x = myScanner.nextInt();
        int y = myScanner.nextInt();
        int z = myScanner.nextInt();
        int w = myScanner.nextInt();

        //assume x is the max for now
        int max = x;

        //if y is greater than our current max, update max
        if (y > max) {
            max = y;
        }

        //if z is greater than our current max, update max
        if (z > max) {
            max = z;
        }

        //if w is greater than our current max, update max
        if (w > max) {
            max = w;
        }

        //Print out the result
        System.out.println("The maximum number is " + max);
    }
}
```

Possible solution to Exercise 3

```
public class Exercise3 {
    public static void main(String[] args) {
        boolean p = true;
        boolean q = true;

        //Print out p XOR q
        System.out.println((p || q) && !(p && q));
    }
}
```