ITI 1120
Lab #4

Branching and some loops

Boolean Expressions

- Evaluate to true or false

Math	Java
AND	`&&`
OR	`||`
NOT	`!`
A = B	`A == B`
A ≤ B	`A <= B`
A ≥ B	`A >= B`
A ≠ B	`A != B`

Truth Tables

- A TRUTH TABLE for a compound Boolean expression shows the results for all possible combinations of the simple expressions:

x	y	x AND y	x OR y
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

Operator NOT

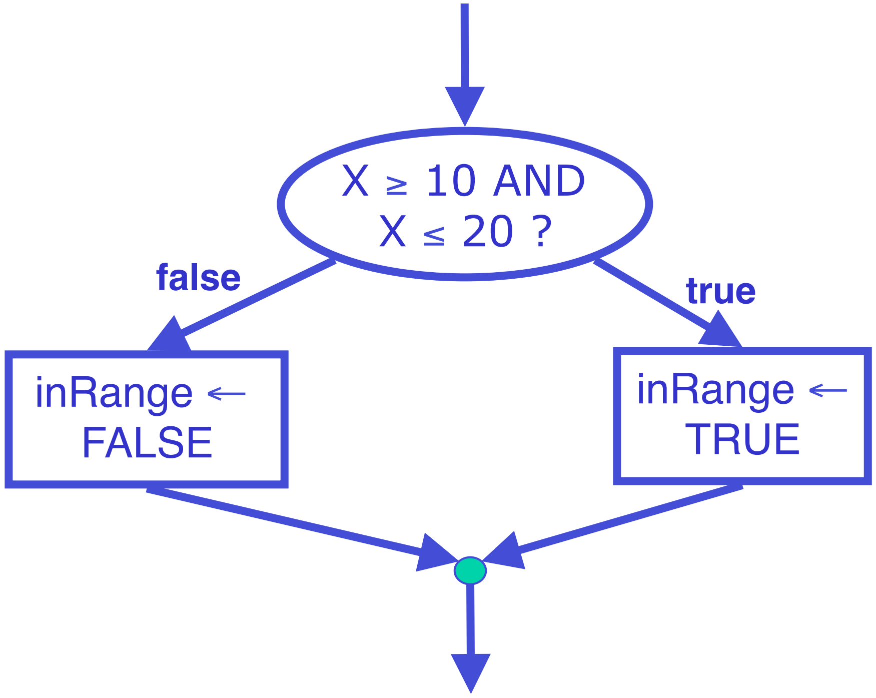
x	NOT x
TRUE	FALSE
FALSE	TRUE

- NOT is an operator to negate the value of a simple or compound Boolean expression:
- Example. Suppose age = 15. Then:
  - Expression age > 16 has a value FALSE, and NOT (age > 16) has a value TRUE.
  - Expression age < 65 has a value TRUE, and NOT (age < 65) has a value FALSE.

Boolean Expressions:

- Write a test that returns TRUE if x is between 10 and 20 (inclusive); the test should return FALSE otherwise

Algorithm:



Java:

```
// assume x has a value
boolean inRange;
if ( (x>=10) && (x<=20) )
{
    inRange = true;
}
else
{
    inRange = false;
}
```

AND versus OR

- In the last slide:
  - We used: `((x>=10) && (x<=20))` to test whether `x` is between 10 and 20.
- What if we used OR `||` instead of AND `&&`
  - Suppose `x` is 7.
  - If we had `((x>=10) || (x<=20))`:

    `x<=20` is TRUE, and so the entire expression is TRUE: but `x` is not between 10 and 20.

Precedence of Operators

- Operators are evaluated left-to-right, with the following precedence (all operators on the same line are treated equally):

  `()` (expression)

  `+ -` (to indicate positive or negative values) `!` (not)

  `* / %`

  `+ -` (for addition or subtraction of two values, concatenation)

  `< > >= <=`

  `== !=`

  `&&`

  `||`

  `=` (assignment to variable)

Operator Precedence

- What is the order of evaluation in the following expressions?

```
a + b + c + d + e            a + b * c - d / e

              a / (b + c) - d % e

              a / (b * (c + (d - e)))
```

Operator Precedence

- What is the order of evaluation in the following expressions?

```
a + b + c + d + e
  1   2   3   4
```

```
a + b * c - d / e
  3   1   4   2
```

```
a / (b + c) - d % e
  2    1    4   3
```

```
a / (b * (c + (d - e)))
  2    3    2    1
```

Exercise 1

```
int i = 10, j = 15, k = 20;
double x = 10.0, y = 2.5, z = 100.0;
```

What are the results of the following 7 boolean expressions? Do this on paper first. Only after you are done, write a program to check your answers.

1. i < j || j < k && x <= y
2. !(j - i < 3) && j % 12 == 3
3. (i / 4) == y
4. (x / 4) == y
5. !(x != i)
6. 'a' != 'b'-1
7. !(!!false || !!true)

Exercise 2

What is the type and value of each of the following expressions in Java? Do this on paper. Only after you are done, write a program to check your answers

Expression	Type	Value
13 * 0.1		
(int) 13 * 0.1		
13 * (int) 0.1		
(int) (13 * 0.1)		
13 % 7		
2<3== 4<5		

Exercise 3:

Write a program that has a main method and a method called isDivisible.

- The method isDivisible takes two integers, n and m as input paramters and returns true if n is divisible by m and false otherwise.
- The **main** method should interact with the user to get two integers and determine if the 1st is divisible by the 2nd by calling isDivisible method. It should print a message explaining the result.

Exercise 4

- Write a program that checks if an integer is divisible by 2 and 3, or just one of 2 or 3, or neither 2 nor 3. The program should have the two methods:
  - The `main` method should interact with the user to get the value of the integer and to display in which of the above three categories it belongs.
  - The `isDivisible23` method has one input parameter, the value of the number. It should return an integer representing if the number is divisible by 2 and 3, divisible by one of 2 or 3, or not divisible by either.

Exercise 5

- The code below is supposed to print the integers from 10 to 1 backwards.
  - You need to find 2 logical errors in the code.
  - Take the time to follow the logic and find the errors BEFORE any coding.
  - Correct the code and insert it into a main method to check out your answer.

```
count = 10;
 while (count >= 0)
 {
  System.out.println(count);
  count = count + 1;
 }
```

Exercise 6

Write a program that has a main method and a method called sumOfSquares. The sumOfSquares method takes integer n as a input parameter and it computes the value of the following series: 1 + 2^2 + 3^2 + 4^2 + ... + n^2, and it returns that value. The main method should prompt the user to input the value for n, and then it should call sumofSquares method by passing that value. Finally it should print the result returned by the sumofSquares method.

Exercise 7 (a bit harder)

(*Game: lottery*) Write a program that lets the user guess two-digit number. A program randomly generates an integer from 10 to 99 for the lottery number. The program prompts the user to enter a two-digit number and determines whether the user wins according to the following rule:

- If the user's guess matches the lottery number, the award is $1,000. (eg. User enters 23 and lottery num is 23)
- If all the digits in the user input match all the digits in the lottery (but the numbers are not the same), the award is $300. (eg. User enters 25 and the lottery num is 52)
- If one digit in the user input matches a digit in the lottery, the award is $100. (e.g. User enters 23 and lottery num is 30)
- Otherwise the user gets nothing

CONTINUE ON THE NEXT PAGE

Examples:

Your program: Enter your lottery pick (two digits):
User: 23
Your program: Lottery is 32
Your program: Match all digits: you win $300

Your program: Enter your lottery pick (two digits):
User: 50
Your program: Lottery is 25
Your program: Match one digit: you win $100

Recall that, Java library has a method Math.random() that generates a random double value in the range [0.0,1.0) (that is greater than or equal to 0.0 and less than 1.0). To generate a random integer value in the range [a,b] (that is, greater than or equal to the value of a or less than or equal to the value of b), you should type-cast the double value to an int value as follows:

```
randomNum = a + (int)(Math.random() * ((b - a) + 1));
```