CS620c Structured Programming Lesson 8

Joe Duffin

Email: Joseph.Duffin@nuim.ie

Selection Statements

- All but the most trivial computer programs need to make decisions.
- They test a condition and operate differently based on the outcome of the test.
- Yesterday we saw:
  - if condition
  - if else condition

if ..else if.. else statement

```
if(condition){
        statements;
           :
}
else if(condition2){
        statements;
           :
}
else {
        statements;
           :
}
```

- `IfElseDemo.java (found in the code examples package)`
- Difference between if and if..else if…else: `If.java IfElse.java`

Switch Statement

- Switch takes as an argument an integer (or character ‘a’, ‘b’ etc) and then checks this value against each of the cases.

```
int x=3;
switch(x)
{
 case 1  : System.out.println(“Monday”);    break;
 case 2  : System.out.println(“Tuesday”);   break;
 case 3  : System.out.println(“Wednesday”); break;
 case 4  : System.out.println(“Thursday”);  break;
 case 5  : System.out.println(“Friday”);    break;
 case 6  : System.out.println(“Saturday”);  break;
 case 7  : System.out.println(“Sunday”);    break;
 default : System.out.println(“Not a day”); break;
}

 Run time display: Wednesday
```

Ternary Operator

- The ternary operator, ?, takes three arguments:
  - a condition, a true value and a false value.
  - It tests the condition and then returns one of two values to the variable based on the result of the condition .

```
y = (boolean_expression) ? true_value : false_value;

int x=-10;
int y=0;


y = (x>0) ? 1 : -1;


System.out.println(y);


Run time display: -1
```

Relational Operators

Operator	Result
==	equal to
!=	not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

Boolean Operators

Operator	**Result**
&	Logical AND
|	Logical OR
^	Logical XOR

We will look at the following next week:

||	Short-circuit OR
&&	Short-circuit AND
!	Logical NOT

Boolean Operators

Operator	**Description**
&	Logical AND
|	Logical OR
^	Logical XOR

AND

A	B	A & B
True	True	True
True	False	False
False	True	False
False	False	False

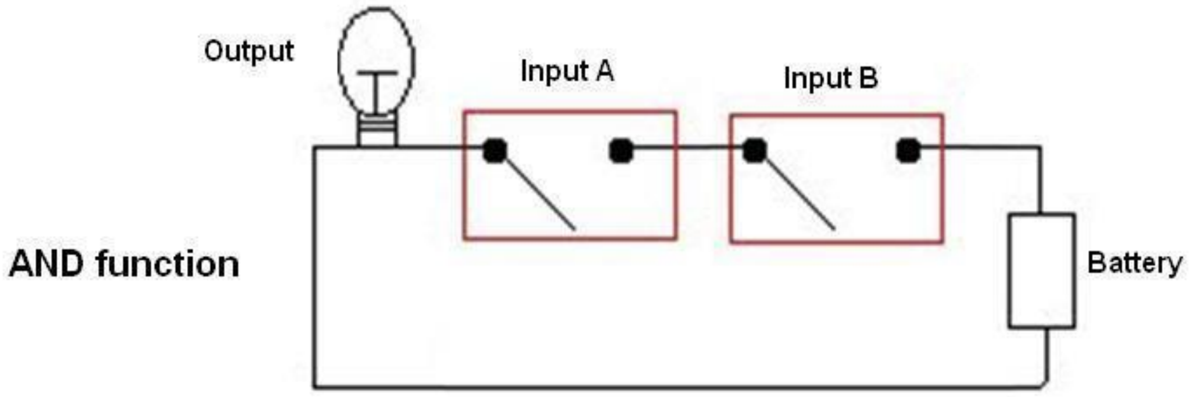
OR

A	B	A | B
True	True	True
True	False	True
False	True	True
False	False	False

Exclusive XOR

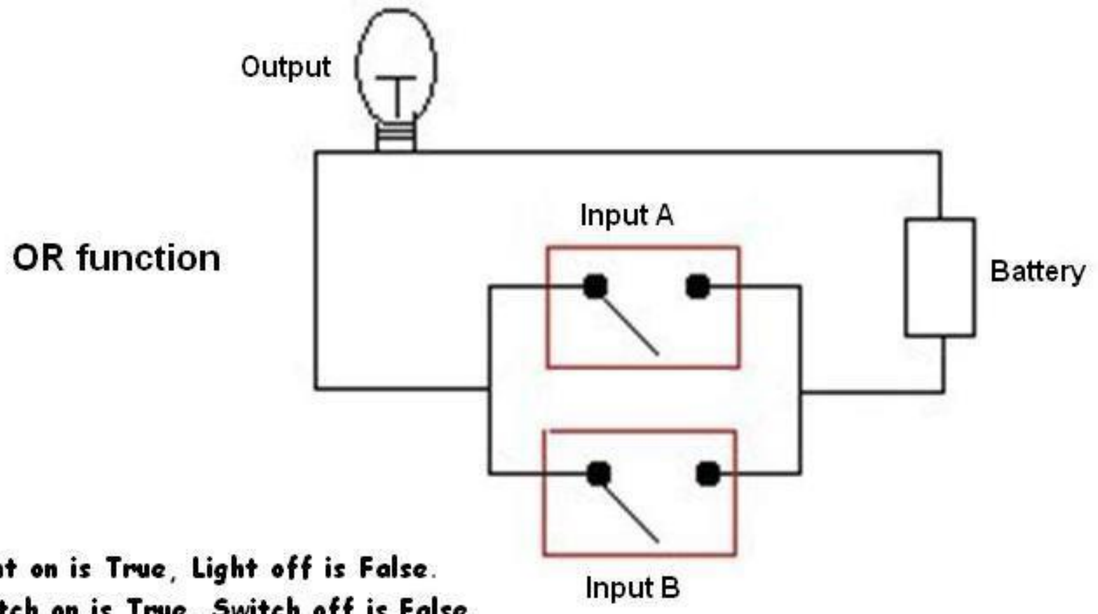
A	B	A ^ B
True	True	False
True	False	True
False	True	True
False	False	False

Electronic Circuits for the Logical AND and the Logical OR functions.



AND (Truth Table)

Input	Input	Output
A	B	A & B
True	True	True
True	False	False
False	True	False
False	False	False



OR (Truth Table)

Input	Input	Output
A	B	A | B
True	True	True
True	False	True
False	True	True
False	False	False

Light on is True, Light off is False.
Switch on is True, Switch off is False.

Boolean Operators

- **Let num1 = 5, num2=7, num3 =5**

If (num1 == 5& num2==5) // True or false?

If (num1 == 5& num3==5)

If (num1 == 5| num2==5)

If (num1 == 5| num3==5)

If (num1 == 5^ num2==5)

If (num1 == 5^ num3==5)

Question for you..

- int result = 1 - 2 * 3 - 4 + 5;
- What value is result after this line executes

Operator Precedence

Highest	( )	[ ]	.		left to right
	++	--	!		right to left
	*	/	%		left to right
	+	-			left to right
	>	>=	<	<=	left to right
	==	!=			left to right
	&				left to right
	^				left to right
	|				left to right
	&&				left to right
	||				left to right
Lowest	=				right to left

- `OperatorPrecedence.java`

Most important!

- Brackets in an expression take precedence
- Followed by increment and decrement operations
- Followed by multiplication, division and modulus
- Followed by addition and subtraction