SPM 2102
PROGRAMMING LANGUAGE 1

Introduction to C++ - Part 2 (Operators)

By

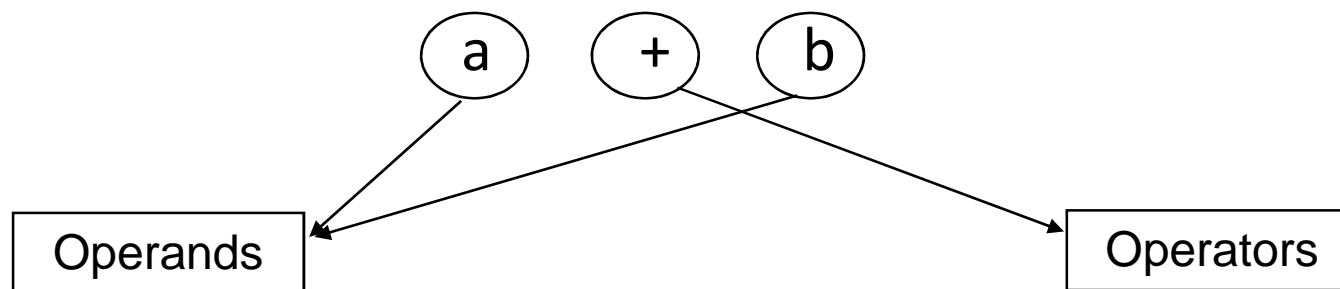NORAH MD NOOR &

MEGAT AMAN ZAHIRI MEGAT ZAKARIA

Topics

- Aritmetic operators

- Comparison operators

- Logical operators

Introduction of expression

- An expression is a sequence of operands and operators that reduce to a single value.

- Example:

$$a \quad + \quad b$$

| Operands | | Operators |

Introduction of expression

- C++ involves a few type of operators in creating expression:

    – Input/Output (eg : cout<< | cin>>)
    – Aritmetic (eg : +, -, *, /, %)
    – Increment & decrement (++ , - -)
    – Comparison (< , ≤ , > , ≥ , ==)
    – Logical

Arithmetic Operators

- What it arithmetic operators?

  – Mathematics basic operators

- Why we need it?

  – Program can perform mathematical calculation

Addition

```cpp
#include <iostream.h>
#include <conio.h>
void main()

{
    float no1, no2, jum;

    cout<<"**ATURCARA PENAMBAHAN DUA INTEGER**\n";
    cout<<"Masukkan nombor pertama: ";
    cin>>no1;
    cout<<"Masukkan nombor kedua: ";
    cin>>no2;

    jum = no1 + no2;

    cout<<"Hasil tambah dua nombor anda ialah: "<<jum;
    cout<<"\nTekan mana-mana kekunci untuk teruskan...";
    getch();
}
```
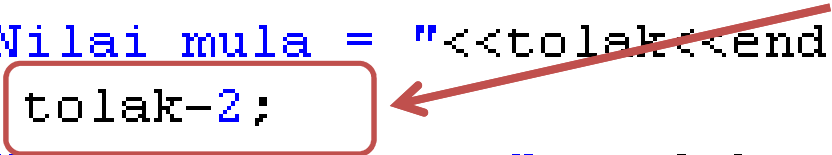
Subtraction

```cpp
#include <iostream.h>
#include <conio.h>
void main()

{

    int tolak;

    tolak = 12;
    cout<<"Nilai mula = "<<tolak<<endl;
    tolak = tolak-2;
    cout<<"Nilai kedua   = "<<tolak<<endl;
    tolak = tolak-2;
    cout<<"Nilai ketiga  = "<<tolak<<endl;
    tolak = tolak-2;
    cout<<"Nilai keempat   = "<<tolak<<endl;
    getch();
}
```
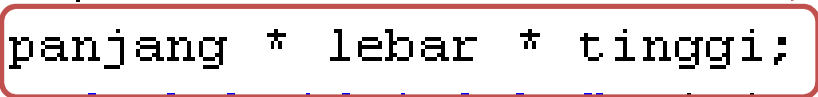
Multiplication

```cpp
#include <iostream.h>
#include <conio.h>
void main()

{
    float panjang, lebar, tinggi, isipadu;

    panjang = 4.5;
    lebar = 3.0;
    tinggi = 2.5;
    isipadu = panjang * lebar * tinggi;
    cout<<"Isipadu kuboid ialah "<<isipadu;

    getch();
}
```

Division

```
#include <iostream.h>
#include <conio.h>
void main()

{
    float  a=6, b=3, jumlah;
    float  purata;

    jumlah = a + b;
    purata = jumlah/2;
    cout<<"Purata ialah "<<purata<<endl;

    getch();
}
```

Modulus

```cpp
#include <iostream.h>
#include <conio.h>

void main()
{
    int no1, no2, bahagi, modulus;

    cout<<"ATURCARA PENGGUNAAN MODULUS";
    cout<<"\n\n\tMasukkan nombor pertama: ";
    cin>>no1;
    cout<<"Masukkan nombor kedua: ";
    cin>>no2;
    bahagi = no1 / no2;
    modulus = no1 % no2;
    cout<<"Hasil bahagi dua nombor anda ialah "<<bahagi;
    cout<<"Baki hasil bahagi tersebut ialah "<<modulus;
    cout<<"\nTekan mana-mana kekunci untuk teruskan…";
    getch();
}
```

Arithmetic: Increment

- Unary increment operator: ++
- x = x + 1; can be written as :
    - ++x; "pre-increment operator"
    - x++; "post-increment operator"
- y = ++x;   is equivalent to
    - x = x + 1 ;
      y = x;
- y = x++;   is equivalent to
    - y = x;
      x = x + 1;

The variable must be an **int**

Arithmetic: Decrement

- Unary decrement operator: --
- x = x - 1; can be written as :
  - --x; "pre-decrement operator"
  - x++; "post-decrement operator"
- y = --x;   is equivalent to
  - x = x - 1 ;
    y = x;
- y = x--;   is equivalent to
 - y = x;
    x = x - 1;

The variable must be an **int**
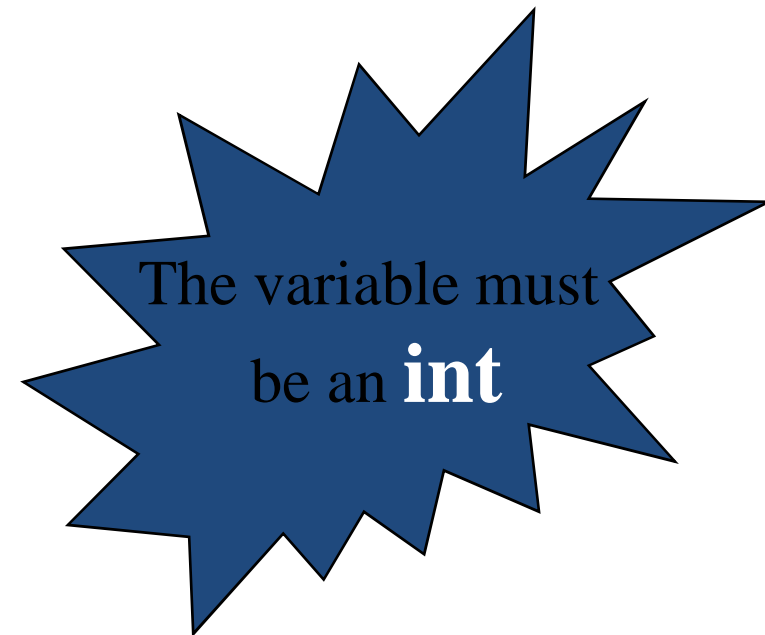
Comparison Operators

- What it comparison operators?

  – Operators use in decision making.

- Why we need it?

  – For logical structure (e.g. If, if-else etc)

- Condition in if structures can be formed by using comparison operators.

Comparison Operators

Operator	Meaning
==	Equality; true if the left-hand argument has the same value as the right
!=	Inequality; opposite of equality
>, <	Greater than, less than; true if the left-hand argument is greater than or less than the right-hand argument
>=, <=	Greater than or equal to, less than or equal to; true if either > or == is true, OR either < or == is true

```
#include <iostream.h>
#include <conio.h>
void main ()
{
short a=2, b=6, c;
 cout<<" ####### \n";
 cout<< (a>b)<<"\n";
 cout<< (a==2)<<"\n";
 getch();
 }
```
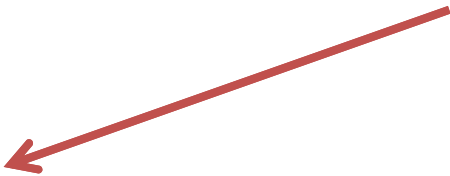
Logical Operators

- What it logical operators?
  - Is it also use in decision making

&&	AND; true if both the left-and right-hand arguments are true
||	OR; true if either the left-or the right-hand argument is true
!	NOT; true if its argument is false

Logical Operators

Truth Table

A	B		A && B	A || B	!A
0	0		0	0	1
0	1		0	1	1
1	0		0	1	0
1	1		1	1	0