

# SPM2102

# BAHASA PENGATURCARAAN 1

Tajuk:

**ELEMEN-ELEMEN DAN PENGENDALI  
DALAM BAHASA C++**

Disediakan oleh:

**NORASYKIN MOHD ZAID  
NORAH MD NOOR**



# Objektif

Di akhir pengajaran ini anda akan mempelajari:

- Elemen-elemen C++
- Pengendali / Operator C++

```
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

**C/C++**

# Elemen-elemen C++

- Elemen-elemen asas dalam C++ terdiri daripada:
  - Set-set aksara
  - Kata kunci
  - Pencam/Pengecam
  - Pemalar
  - Rentetan
  - Penanda
  - Pengendali

# Elemen-elemen C++

- **Set-set aksara**
  - Blok binaan asas dalam aturcara C++
  - Set aksara terdiri daripada 92 aksara
    - Huruf besar A hingga Z (UpperCase)
    - Huruf kecil a hingga z (LowerCase)
    - Nombor 0 hingga 9
    - Set aksara dalam C++ terdiri daripada aksara-aksara yang terdapat di atas sesebuah papan kekunci yang piawai kecuali aksara ` , \$ dan @
    - Aksara kosong atau ruang kosong

- Aksara-aksara lain
  - Punctuation symbols , . : ; ? !
  - Parentheses ( )
  - Braces { }
  - Brackets [ ]
  - Quotation marks “ ‘
  - Arithmetics symbols + - \* /
  - Comparison symbols = ><
  - Pound #, percent %, ampersand &, caret ^, tilda ~, vertical slash |, backslash \, underscore \_

# Elemen-elemen C++

- Aksara boleh digabungkan mengikut peraturan sintak untuk membentuk **token**
- Token merupakan unsur bahasa yang boleh digunakan untuk membentuk bahasa peringkat tinggi. Terdapat 6 jenis token dalam C++

# Elemen-elemen C++

- Terdapat enam jenis token dalam bahasa C++:
  - Kata kunci *reserved words/keywords*
  - Pencam/pengecam *identifiers/user-defined words*
  - Pemalar *constants*
  - Rentetan *literal string*
  - Penanda *punctuators/separator*
  - Pengendali *operator*

# Elemen-elemen C++

- **Kata kunci**
  - Kata kunci merupakan **perkataan-perkataan simpanan** dan tidak boleh digunakan sebagai pemanggilan yang ditakrifkan oleh pengguna
  - mempunyai makna tertentu kepada pengkompil
  - Terdapat 32 kata kunci:
  - auto, extern, sizeof, break, float, static, for, struct, char, goto, switch, const, if, typedef, continue, int, union, default, long, unsigned, do, register, void, double, return, volatile, else, short, while, enum, signed, case, main



# Elemen-elemen C++

- **Pencam/pegecam**
  - Nama yang diberikan untuk pelbagai unsur di dalam aturcara. Cth:nama pembolehubah
  - Perkataan ditakrifkan oleh pengguna
  - Nama pencam mesti unik
  - Peraturan untuk menamakan pencam
    - Aksara pertama mesti huruf atau garis bawah
    - Boleh terdiri daripada A-Z, a-z, 0-9 dan garis bawah \_
    - Tiada had panjang tetapi pengkompil akan mengenali 32 aksara pertama
    - Tiada tempat kosong
    - Kata kunci tidak boleh digunakan
    - Case sensitive, cth: **Jumlah** tidak sama dengan **jumlah**

# Elemen-elemen C++

- Contoh pemalar yang sah:
  - `nama_pelajar data01 namaPelajar`
- Contoh pemalar tidak sah:
  - `Nama pelajar 01data int`
- **Pemalar**
  - Suatu entiti yang terkandung di dalam aturcara dengan nilai tetap
  - Terdapat beberapa jenis pemalar:
    - Pemalar integer
    - Pemalar titik apung
    - Pemalar aksara

# Elemen-elemen C++

- **Pemalar integer**
  - Satu jujukan digit yang boleh ditulis dengan menggunakan sistem penomboran yang berbeza tanpa mengandungi nilai pecahan.
- Pemalar integer boleh menggunakan sistem penomboran desimal (base 10), oktal (base 8) dan hesadesimal (base 16)

# Elemen-elemen C++

Peraturan membina pemalar integer:

- a. Mulakan dengan digit desimal bukan sifar
- b. Jika tiada tanda – atau +, maka sistem komputer akan mengandaikan positif
- c. Comma tidak dibenarkan dalam pembinaan pemalar integer

cth: 100, 00 (tidak betul), 100 00 (betul)

contoh pemalar integer: -15, +250 dan 7550

# Elemen-elemen C++

- **Pemalar titik terapung**

- Terdiri dari nombor perpuluhan atau eksponen dengan nilai negatif/positif.

Contoh : 10.0,  $0.1 \times 10^2$ , 0.1E+2, 0.1e+2

- 0.0 mengikut notasi konvensional
- $0.1 \times 10^2$  mengikut notasi saintifik
- 0.1E+2 dan 0.1e+2 mengikut notasi yang ditulis dalam C++ (e/E merujuk kepada eksponen)

# Elemen-elemen C++

- **Pemalar aksara**
- - Terdiri dari satu aksara di dalam ' ' dan ianya mempunyai nilai.

Contoh : 'A' 'x' '2' '%'

```
#DEFINE maxint 32767  
atau:  
const maxint = 32767;
```

# Elemen-elemen C++

- **Escape Sequence**

- Terdapat aksara-aksara khas yang bermula dengan \.
- Aksara ini dikenali sebagai aksara lepasan (*escape sequence*) (tidak boleh nampak)
- cth: \n mewakili baris baru (new line)
  - \t mewakili tab
  - \” mewakili aksara pembuka kata berganda
  - \0 mewakili aksara null
  - \’ mewakili aksara pembuka kata tunggal

## Operator-operator 'escape' di dalam C++:

Escape Sequence	Description
<code>\n</code>	Newline. Position the screen cursor to the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the screen cursor to the next tab stop.
<code>\r</code>	Carriage return. Position the screen cursor to the beginning of the current line; do not advance to the next line.
<code>\a</code>	Alert. Sound the system bell.
<code>\\</code>	Backslash. Used to print a backslash character.
<code>\"</code>	Double quote. Used to print a double quote character.



```
1 // Fig. 1.4: fig01_04.cpp
2 // Printing a line with multiple statements
3 #include <iostream>
4
5 int main()
6 {
7     cout << "Welcome ";
8     cout << "to C++!\n";
9
10     return 0; // indicate that program ended
11 }
```

Welcome to C++!

Unless new line '`\n`' is specified, the text continues on the same line.

```
1 // Fig. 1.5: fig01_05.cpp
2 // Printing multiple lines with a single statement
3 #include <iostream>
4
5 int main()
6 {
7     cout << "Welcome\nto\n\nC++!\n";
8
9     return 0; // indicate that program ended successfully
10 }
```

```
Welcome
to
C++!
```



Multiple lines can be printed with one statement.

# Elemen-elemen C++

- **Rentetan**

- Terdiri dari aksara di dalam " "
- Berbeza dengan pemalar aksara, pemalar rentetan tidak mengandungi nilai

Contoh : "hijau" "Beverly Hills" "\$2.50"

- **Penanda**

- Penanda atau pemisah digunakan dengan banyak dalam aturcara C++

Contoh : [] () {} , ; :

# Pengendali / Operator

- Pengendali atau Operator adalah token yang digunakan untuk menghasilkan **hasil pengiraan** atau **tindakan**
- Terdapat 6 jenis pengendali:
  - Pengendali input/output
  - Pengendali aritmetik
  - Pengendali Penokokan dan Penyusutan
  - Pengendali Umpukan
  - Pengendali Hubungan/Pengendali Logik

# Pengendali / Operator

- **Operator Aritmetik dan penokokan serta penyusutan**

Operator	Penerangan
-	Operasi tolak
+	Operasi tambah
*	Operasi darab
/	Operasi bahagi
%	Operasi modulus (memulangkan baki dalam bentuk integer)
--	Pengurangan
++	penambahan

Penokokan dan penyusutan

# Pengendali / Operator

- Peraturan keutamaan operator

Operator(s)	Operation(s)	Order of evaluation (precedence)
()	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses “on the same level” (i.e., not nested), they are evaluated left to right.
*, / , %	Multiplication, Division , Modulus	Evaluated second. If there are several, they re evaluated left to right
+ or -	Addition or substraction	Evaluated last. If there are several, they are evaluated left to right

# Pengendali / Operator

- **Operator Hubungan dan logik**

Operator Hubungan	Penerangan
>	Lebih besar daripada
>=	Lebih besar atau sama dengan
<	Lebih kecil daripada
<=	Lebih kecil atau sama dengan
==	Sama dengan
!=	Tidak sama dengan
<b>Operator logik</b>	
&&	AND
	OR
!	NOT

# Pengendali / Operator

- && (logical AND)
  - Memulangkan true sekiranya kedua-dua kondisi adalah true (benar)
- || (logical OR)
  - Memulangkan true sekiranya salah satu drp kondisi adalah true
- ! (logical NOT, logical negation)
  - Memulangkan true apabila kondisi adalah false/palsu, dan sebaliknya
  - Merupakan operator unari, iaitu ia hanya mengambil satu kondisi sahaja



## Operator hubungan (relational) & persamaan (equality)

- Berguna untuk membandingkan perhubungan antara dua pernyataan.
- Operator hubungan dan persamaan
  - Mempunyai keutamaan yg lebih rendah drp operator aritmetik
- Hasilnya adalah **BETUL (1)** atau **SALAH (0)**

<	lebih kecil
<=	lebih kecil atau sama
>	lebih besar
>=	lebih besar atau sama
==	sama
!=	tak sama

## Operator hubungan (relational) & persamaan (equality)

- *contoh:*
- **a = 1; b = 2; c = d = 3;**

<code>return ( c &lt;= a + b )</code>	<code>return (1)</code>
<code>return ( a &gt; b )</code>	<code>return (0)</code>
<code>return ( d == a + b )</code>	<code>return (1)</code>

- **c <= a + b** adalah setara dengan **c <= ( a + b )**

Expression 1	Expression 2	expression1 && expression2
false	false	false
false	true	false
true	false	false
true	true	true

**Fig. 2.28** Truth table for the && (logical AND) operator.

Expression 1	Expression 2	Expression1 expression 2
false	false	false
false	true	true
true	false	true
true	true	true

**Fig. 2.29** Truth table for the | (logical OR) operator.

# Pengendali / Operator

- **Pengendali kawalan /umpukan**

Apabila sesuatu pembolehubah diberi nilai yang tertentu diawal aturcara ia dipanggil pengawalan atau initialization.

```
int nombor = 1;  
float peratus = 0.5;  
char gred = 'A';  
int a = x + y;
```

# Pengendali / Operator

Umpukan (assignment) dilakukan sama seperti pememulaan tetapi tanpa kata kunci jenisnya. Jadi contoh-contoh berikut adalah untuk umpukan:

```
nombor = 2; gred = 'B';  
purata = jumlah / bilangan_subjek;  
a = x + y;
```

# Pengendali / Operator

- Operator assignment/umpukan boleh diringkaskan
  - $c = c + 3$ ; boleh diringkaskan kepada  $c += 3$ ;
- Secara umumnya, pernyataan dlm btk  
Pembolehubah = pembolehubah operator  
expression;  
boleh diringkaskan kepada  
pembolehubah operator= expression;

# Pengendali / Operator

- Contoh:-

**d -= 4** (d = d - 4)

**e \*= 5** (e = e \* 5)

**f /= 3** (f = f / 3)

**g %= 9** (g = g % 9)

# Pengendali / Operator

- **Pengendali Input/Output**
- Set jenis dan rutin yang terdapat dalam perpustakaan piawai (standard library).
- Perpustakaan piawai C++ bagi I/O ialah **iostream.h**
  - << aliran **output** dianjak ke kiri
  - >> aliran **input** dianjak ke kanan
- Pengisytiharan tiga aliran piawai :
  - cout //piawai keluaran
  - cin //piawai input
  - cerr //piawai ralat



# Pengendali / Operator

- `cout`
  - Merupakan operator penyelitan jujukan (stream insertion operator), dimana nilai pada sebelah kanan operator `<<` akan diselitkan kedalam jujukan output
- `cout` bersama `<<` digunakan utk mencetak jujukan aksara yang terkandung di antara “ dan “ ke atas skrin
- Contoh: `cout << “Welcome to C++!\n”;`

akan mencetak *Welcome to C++!* di atas skrin diikuti dgn selang sebaris.

# Pengendali / Operator

- cin
  - Merupakan piawaian bagi input
- >>
  - ia akan menunggu pengguna memasukkan satu nilai input dan kemudian menyimpan nilai tersebut ke dalam pembolehubah di sebelah kanan operator >>
  - Pengguna perlu memasukkan nilai dan menekan kekunci 'Enter' sebelum data dapat dihantar ke komputer
- Contoh: -

```
int myVariable;
```

```
cin >> myVariable;
```

```
//tunggu pengguna memasukan nilai, kemudian menyimpan  
nilai
```

```
//tersebut di dalam myVariable
```

```
1 // Fig. 1.6: fig01 06.cpp
2 // Addition program
3 #include <iostream>
4
5 int main()
6 {
7     int integer1, integer2, sum;           // declaration
8
9     cout << "Enter first integer\n"; // prompt
10    cin >> integer1;                     // read an integer
11    cout << "Enter second integer\n"; // prompt
12    cin >> integer2;                     // read an integer
13    sum = integer1 + integer2;           // assignment of sum
14    cout << "Sum is " << sum << endl; // print sum
15
16    return 0; // indicate that program ended successfully
17 }
```

Notice how **cin** is used to get user input.

```
Enter first integer
45
Enter second integer
72
Sum is 117
```

Variables can be output using **cout**  
**<< variableName.**

**endl** flushes the buffer and prints a newline.

# Rujukan

- Jabatan Multimedia Pendidikan (2004), *Pengenalan kepada Bahasa Pengaturcaraan dalam Konteks Bahasa C++* , Faculti Pendidikan, UTM ( unpublished )
- Abdullah Zawawi Talib, Ahamad Tajudin Khader, Maziani Sabudin, Wahidah Husain (2002), *Prinsip-Prinsip Pengaturcaraan Menggunakan C++ : Edisi kedua*, Pearson Malaysia, ISBN 983-9236-87-3
- Bakar, M. A et al (2002). Pengaturcaraan C. Malaysia. Prentice Hall
- Deitel, P. J., & Deitel, H. M. (2005). *C++ How to Program (5th Edition)*, Prentice Hall