مدخل إلى الخوارزميات والبرمجة هندسة الميكاترونيكس سنة أولى

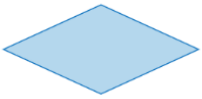
مدرس المقرر: د.عيسى الغنام 2023-2024

Lecture No.3

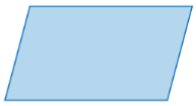
Part 1- Pseudocode and Flow chart

Start/End Symbol
The terminator symbol marks the starting or ending point of the system. It usually contains the word "Start" or "End."

Action or Process Symbol
A box can represent a single step ("add two cups of flour"), or an entire sub-process ("make bread") within a larger process.

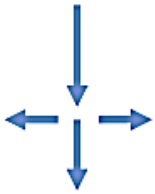


Decision Symbol
A decision or branching point. Lines representing different decisions emerge from different points of the diamond.



Input/Output Symbol
Represents material or information entering or leaving the system, such as customer order (input) or a product (output).

Connector Symbol
Indicates that the flow continues where a matching symbol (containing the same letter) has been placed.



Flow direction

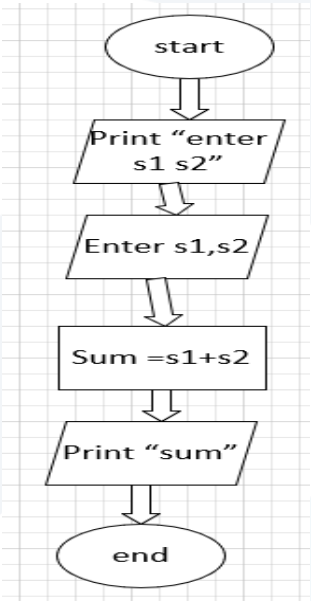
للتحكم باتجاه التدفق

Pseudocode Example 1: Add Two Numbers.

Pseudocode

1. BEGIN
2. NUMBER s1, s2, sum
3. OUTPUT("Input number1:")
4. INPUT s1
5. OUTPUT("Input number2:")
6. INPUT s2
7. sum=s1+s2
8. OUTPUT sum
9. END

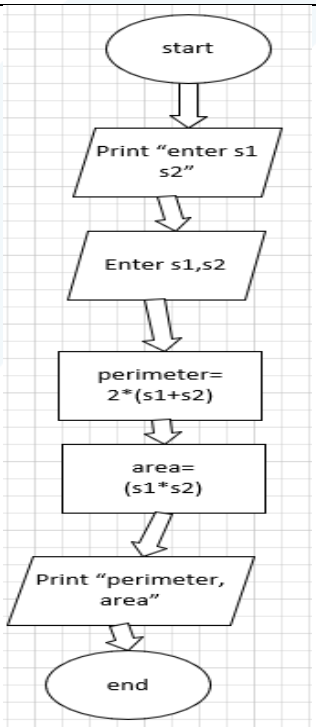
Flowchart




```
#include <iostream>
using namespace std;
int main() {
 int first_number, second_number,
sum;
 cout << "Enter two integers: ";
 cin >> first_number >>
second_number;
 // sum of two numbers in stored in
//variable sumOfTwoNumbers
 sum = first_number + second_number;
 // prints sum
 cout << first_number << " + " <<
second_number << " = " << sum;
 return 0;}
```

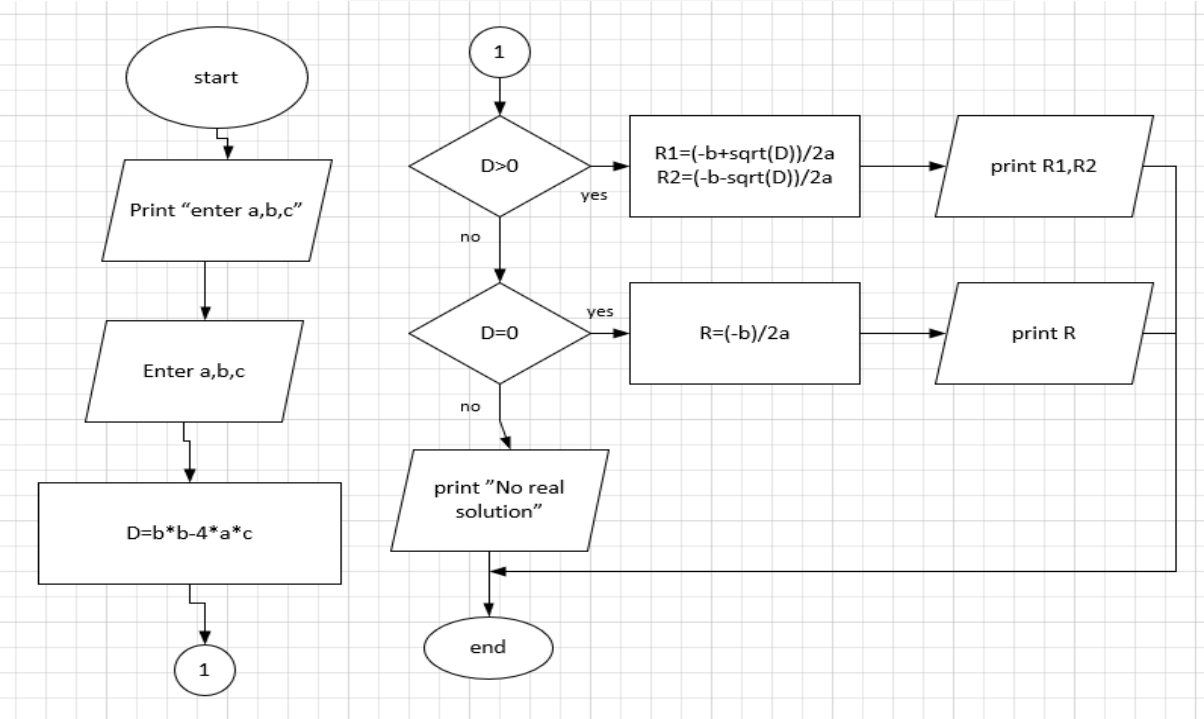
Pseudocode Example 2: Calculate Area and Perimeter of Rectangle

1. BEGIN
2. NUMBER b1,b2,area,perimeter
3. INPUT s1
4. INPUT s2
5. area=s1*s2
6. perimeter=2*(s1+s2)
7. OUTPUT area
8. OUTPUT perimeter
9. END




```
#include <iostream>
using namespace std;
int main(){
int width, lngth, area, peri; //
cout << " Input the length of the rectangle : ";
cin >> lngth;
cout << " Input the width of the rectangle : ";
cin >> width;
area = (lngth * width);
peri = 2 * (lngth + width);
cout << " The area of the rectangle is : "<< area
<< endl;
cout << " The perimeter of the rectangle is : "<<
peri << endl;
cout << endl;
return 0;}
```

Pseudocode If Else Example: Solve Quadratic Equation: ax2+bx+c=0



1. BEGIN
2. NUMBER a, b, c, d, R1, R2
3. INPUT a,b,c
4. d = b^2-4ac
5. IF (d >= 0) THEN
6. R1 = (-b+√d)/2a
7. R2 = (-b-√d)/2a
8. OUTPUT "ROOT 1:"R1
9. OUTPUT "ROOT 2:"R2
10. ELSE IF (d == 0) THEN
11. R1=R2= -b/2a
12. OUTPUT "ROOT 1:"R1
13. ELSE
14. OUTPUT "There is no real root"
15. ENDIF
16. END
	#include <iostream>
	#include <cmath>
	using namespace std;
	int main() {
	float a, b, c, x1, x2, DELTA;
	cout << "Enter coefficients a, b and c: ";
	cin >> a >> b >> c;
	DELTA = b*b - 4*a*c;
	if (DELTA > 0) {
	x1 = (-b + sqrt(DELTA)) / (2*a);
	x2 = (-b - sqrt(DELTA)) / (2*a);
	cout << "Roots are real and different." << endl;
	cout << "x1 = " << x1 << endl;
	cout << "x2 = " << x2 << endl; }
	else if (DELTA == 0) {
	cout << "Roots are real and same." << endl;
	x1 = -b/(2*a);
	cout << "x1 = x2 =" << x1 << endl;}
	else {
	cout << "Roots are complex and different." << endl;}
	return 0;}

Part 2- C++ Operators

2-1 Introduction:

Operators are used to perform operations on variables and values. In the example below, we use the + **operator** to add together two values:

```
int x = 100 + 50;
```

Although the + operator is often used to add together two values, like in the example above, it can also be used to add together a variable and a value, or a variable and another variable:

```
int sum1 = 100 + 50;      // 150   (100 + 50)
int sum2 = sum1 + 250;    // 400   (150 + 250)
int sum3 = sum2 + sum2;   // 800   (400 + 400)
```

C++ divides the operators into the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Bitwise operators

2-2 Arithmetic Operators:

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

Example: المشغلات المختصرة الزيادة والإنقاص السابقة واللاحقة

```
#include <iostream>
using namespace std;
int main()
{

        char c=64;
        cout<<c<<" ";
        cout<<c++<<" ";
        cout<<c++<<" ";
        cout<<c++<<" ";
        cout<<c ++<<endl;

        c =96;
    cout<<c<<" ";
        cout<<c ++<<" ";
        cout<<c ++<<" ";
        cout<<c ++<<" ";
        cout<<c ++<<endl;

        int c1=64;
        cout<<c1<<" ";
        cout<<c1++<<" ";
        cout<<c1++<<" ";
        cout<<c1++<<" ";
        cout<<c1++<<endl;

        int c2=64;
        cout<<c2<<" ";
        cout<<++c2<<" ";
        cout<<++c2<<" ";
        cout<<++c2<<" ";
        cout<<++c2<<endl;

        int c3=64;
        cout<<c3<<" ";
        cout<<c3--<<" ";
        cout<<c3--<<" ";
        cout<<c3--<<" ";
        cout<<c3--<<endl;

        int c4=64;
        cout<<c4<<" ";
        cout<<--c4<<" ";
        cout<<--c4<<" ";
        cout<<--c4<<" ";
        cout<<--c4<<endl;

    int g=0;

cout<<g<<"\t"<<c4<<"\t"<<c2<<endl;
    //g=(c4++) + (--c2);
    g=c4++ + --c2;

cout<<g<<"\t"<<c4<<"\t"<<c2<<endl;
    int g1=0;
cout<<g1<<"\t"<<c4<<"\t"<<c2<<endl;
//g1=(--c4) + (--c2);
g1=--c4 + --c2;
cout<<g1<<"\t"<<c4<<"\t"<<c2<<endl;

        return 0;
}
```

```
"F:\essa\mechatronic\mecha\work\c++\٢٠١٢\٣\١٧\على الطلاعطو
@ @ A B C
` ` a b c
64 64 65 66 67
64 65 66 67 68
64 64 63 62 61
64 63 62 61 60
0        60       68
127      61       67
0        61       67
126      60       66

Process returned 0 (0x0)   execution time : 0.141 s
Press any key to continue.
```

3-2 Assignment Operators

Assignment operators are used to assign values to variables. In the example below, we use the **assignment** operator (=) to assign the value **10** to a variable called **x**:

```
int x = 10;
```

The **addition assignment** operator (+=) adds a value to a variable:

Example

```
int x = 10;
x += 5;
```

A list of all assignment operators:

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Exmaple: الصيغ المختصرة لإسناد المعطيات

```
1.  #include <iostream>
2.
3.  using namespace std;
4.
5.  int main()
6.  {
7.     int x=0,y=5,z=4,k=120;
8.     x+=2;
9.     cout<<"x="<<x<<"\n";
10.    y-=3;
11.    cout<<"y="<<y<<"\n";
12.    z-=y;
13.    cout<<"z="<<z<<"\n";
14.    y*=5;
15.    cout<<"y="<<y<<"\n";
16.    x/=2;
17.    cout<<"x="<<x<<"\n";
18.    k%=100;
19.    cout<<"k="<<k<<"\n";
20.    k/=y;
21.    cout<<"k="<<k<<"\n";
22.    k*=k+x;    //k=k*(k+x)
23.    cout<<"k="<<k<<"\n";
24.    return 0;
25. }
```

```
"E:\0.SCIENCE\0.courses\1.programming languages\c++\0 course\(
x=2
y=2
z=2
y=10
x=1
k=20
k=2
k=6

Process returned 0 (0x0)   execution time : 0.067 s
Press any key to continue.
```

انتهت المحاضرة