المعهد التقني النجف

تقنيات أنظمة الحاسوب

المرحلة الأولى

الخوارزميات

ضُحى عامر مهدي

*تراكيب السلسلة/ تراكيب الاختبار (الشرطية)

هناك نظرية اساسية للبرمجة المهيكلة ترتكز على مبدأ امكانية صياغة البرنامج المضبوط الذي يبدأ بمدخل واحد ومخرج واحد باستخدام ثلاث تراكيب أساسية:

1. Sequence -التتابع او السلسلة
2. Conditionalالشرطية او الاختبار
3. Repetition

وتتمثل هذه التركيب باستخدام المخططات الانسيابية التي مهما كانت درجة تعقيد برامجها فانه يجب ان تكون قابلة للاختزال للتراكيب أعلاه.

نلجأ الى هذه التراكيب من ناحيتين:

1- أي برنامج يكتب يحتاج ان تكون هناك امكانية دون اللجوء الى الاستخدام الواضح للأمر GOTO
2- هيكل كل تركيب سيكون له نقطة دخول (Entry) واحدة ونقطة خروج (Exit) واحدة.

تركيب السلسلة

يشير الى المجرى المنطقي المتتالي للبرامج.

//Program 1: //This program display a message on the screen.

```
#include <iostream>
main ( )
{ cout <<''welcome to C++ !\n'';
return 0; }
```

Program2: Addition Program//

```
#include<iostream>
 main ( )
{ int integer1, integer2, sum;
cout <<"Enter first integer\n";
 cin >> integer1;
cout <<"Enter second integer\n";
 cin >> integer2;
 sum= integer1+integer2;
cout <<"sum="<<sum<endl;
return 0; }
```

تركيب الاختبار

يتعرض مجرى البرنامج تركيب شـــرطي يؤدي الى تفرع البرنامج الى فرعين يجب الاختبار بينهما. يتم اختبار التسلسل المنطقي الملائم تبعا للشرط فإما ان يصبح الشرط False أو True

If condition True then value:=function 1

Else

Value:= function2

وقد تصادف في بعض الاحيان حالات شرطية ابسط يترتب عليها تفرع واحد يتبعه مجرى البرنامج الذي يصحح فيه الشرط والا فان البرنامج يواصل تنفيذه لإنجاز الايعاز التالي.

If condition True then value:=function

Program 3:

هذا البرنامج مكتوب بلغة C++ يوضح IF Statement

```
#include<iostream>
 main ( ) {
 int num1 , num2;
cout << " Enter two integers, and I will tell you\n" <<" the relationships they satisfy: ";
 cin >> num1>> num2;
 if (num1== num2)
 cout << num1 << " is equal to " << num2 << endl;
if (num1!= num2)
 cout << num1 << " is not equal to " << num2 << endl;
 if (num1< num2) cout << num1 << " is less than " << num2 << endl;
 if (num1> num2) cout << num1 << " is greater than " << num2 << endl;
 if (num1<= num2) cout << num1 << " is less than or equal to " << num2 << endl;
 if (num1>= num2) cout << num1 << " is greater than or equal to " << num2 << endl;
return 0; }
```

Program 4:

هذا البرنامج مكتوب بلغة C++ يوضح If-else Statement

```
#include<iostream>
 main ( ) {
 int grade ;
 cout << " Enter the grade";
cin >>grade;
if(grade>= 50)
 cout<<"pass" <<<"fail"<<endl;
```

Program 5:

هذا البرنامج مكتوب بلغة C++ يوضح Nested-If _else Statement

```
#include<iostream>
 main ( ) {
 int grade;
 cout <<"Enter the grade:"
 ; cin >> grade;
 if(grade>= 75)
cout<<'A'<< endl;
 else if(grade>= 65)
 cout<<'B'<< endl;
 else if(grade>= 55)
 cout<<'C'<< endl;
else if(grade>= 40)
 cout<<'D'<< endl;
 else cout<<"fail"<<endl;
return 0; }
```

تركيب التكرار

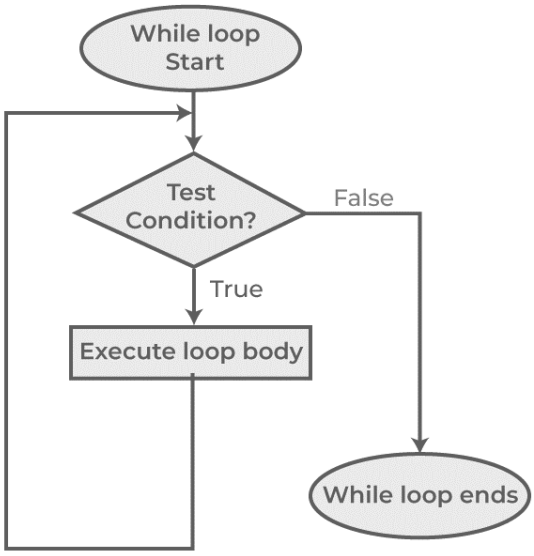
يسمح بعمليات التكرار الدائري Loop أو بعبارة أخرى يسمح بتكرار تنفيذ مجموعة ايعازات عدة مرات.

حلقة التكرار WHILE

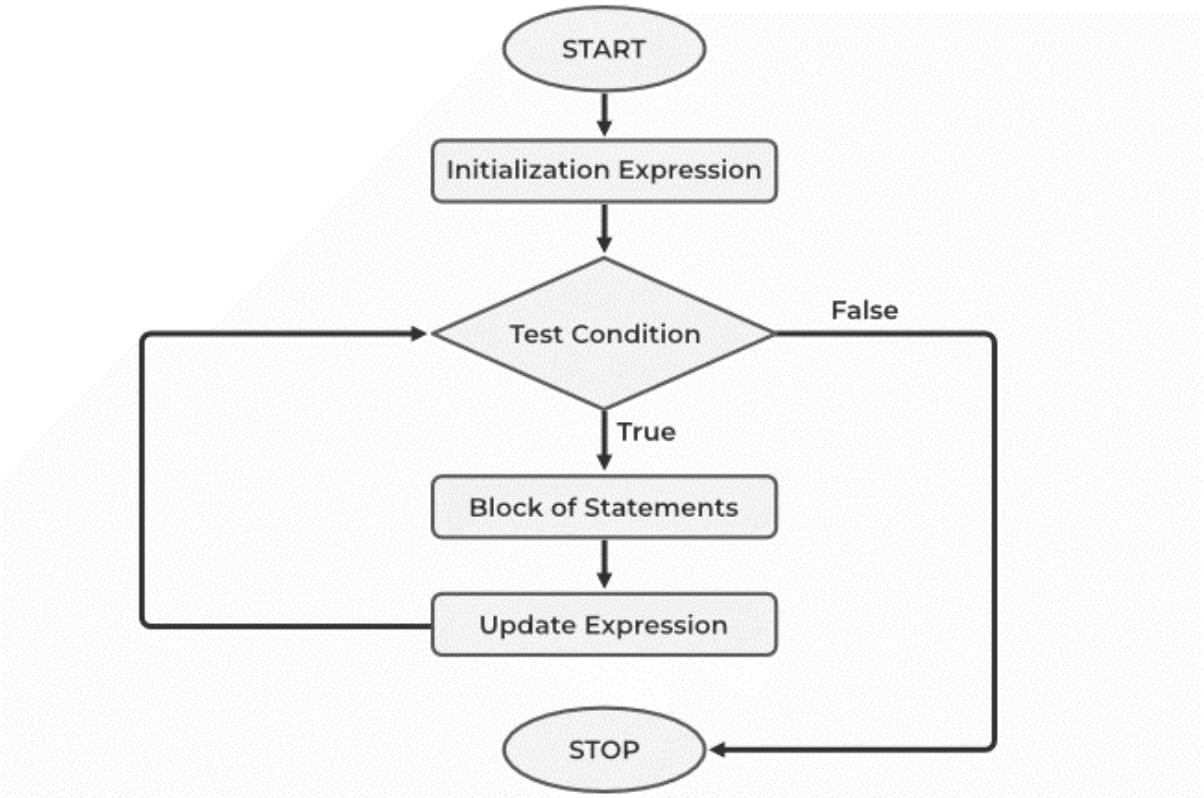
Program 6:

البرنامج التالي بلغة C++ يوضح حلقة التكرار While

```
#include <iostream>
using namespace std;

int main()
{
   int i = 1;

   while (i < 6) {
      cout << "Hello World\n";

      i++;
   }
   return 0;
}
```

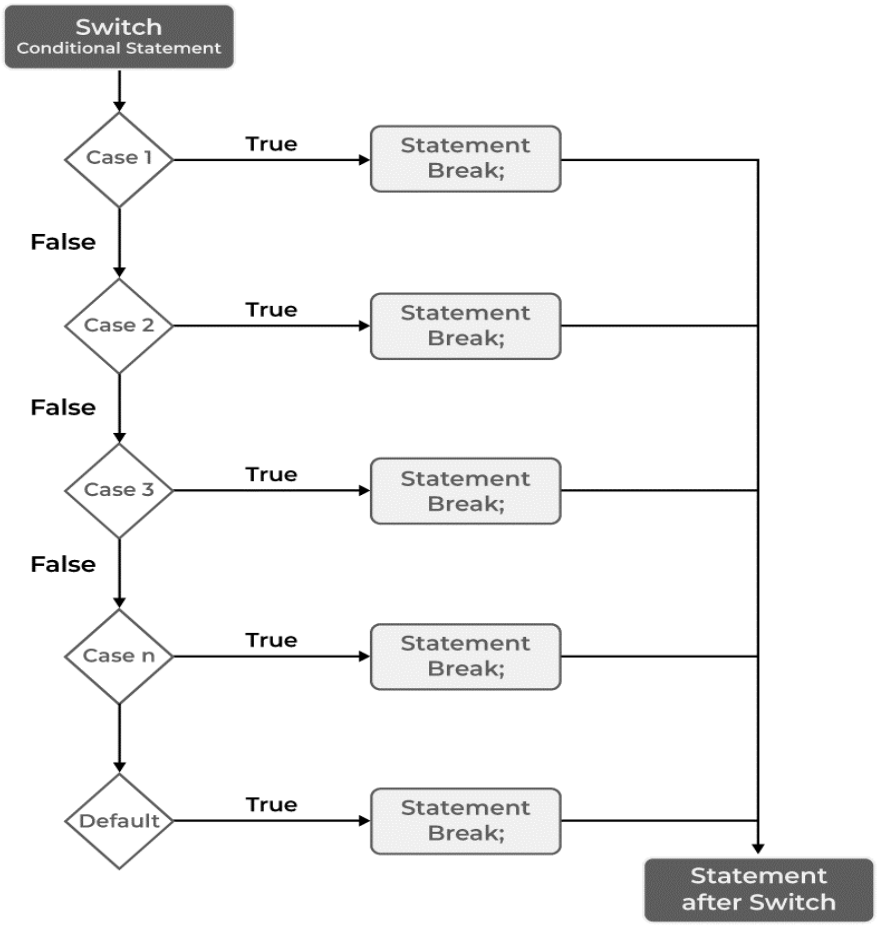


حلقة التكرار FOR



```
#include <iostream>
int main()
{
   // Writing a for loop
   // to print Hello World 5 times
   int i;
   for (i = 1; i <= 5; i++) {
      cout << "Hello World\n";
   }
   return 0;
}
```

حلقة التكرار SWITCH CASE



```
#include <iostream>
using namespace std;
int main()
{
   int var1 = 1;
   int var2 = 0;
   // outer switch
   switch (var1) {
     case 0:
        cout << "Outer Switch Case 0\n";
        break;
     case 1:
        cout << "Outer Switch Case 1\n";
        // inner switch
        switch (var2) {
        case 0:
           cout << "Inner Switch Case 0\n";
           break;
        }
        break;
     default:
        cout << "Default Case of Outer Loop";
        break;
   }
```