Program Documentation

***** Too important to ignore**

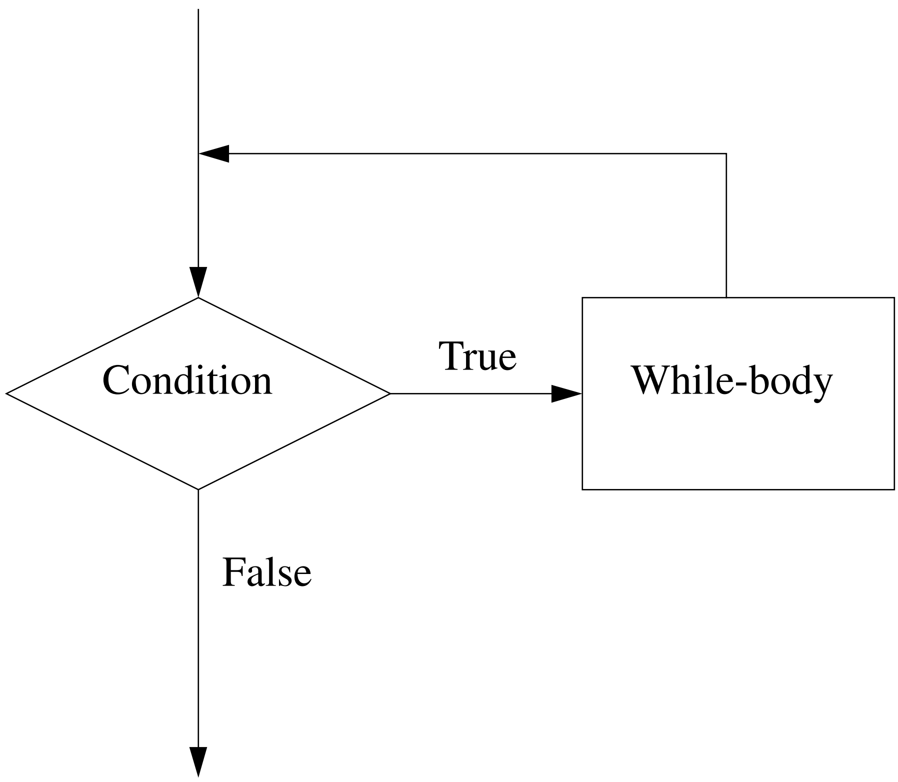
1. Requirement Specification. (Problem Definition)
2. Architecture Specification. (Solution Outline)
3. Detail Design Specification. (Solution Refinement)
4. In-Line Documentation. (Program Documentation)
5. Unit-Test Specification. (Test Cases for each Module)
6. System Test Specification. (Test Cases for whole Program)

WHILE Loop

while (< logical expression >) {

< sequence of statements >

}



While Loop (Example)

```
// Program to find the maximum value of a list of numbers
// Input:    A list of numbers
// Output:  Maximum value of the numbers entered

void main( )
{
int  value,                                   //Input value
    maxval;                                   //Maximum value

     value = 0;                               // Initialize value
     maxval = 0;                              // Initialize maxval

     while  (value != -1) {
       cout << "Enter a value (-1 to stop): ";
       cin >> value;

       if (value > maxval)
          maxval = value;                     // new maximum
     }

     cout << "The maximum value found is "
         << maxval << endl;
}
```

While Loop (Example)

```
// Program to find the sum of numbers from 1 to n
// Input:      n
// Output:    sum of numbers from 1 to n

#include <iostream.h>

void main ( ) {
    int   n,       // Input number
          sum,       //  Initialize sum to 0
           i;

    cout << “Enter a positive number: “;    cin >> n;
    i = 0;
    sum = 0;

    while (i  <=  n) {
        sum = sum + i ;
        i++  ;                          //  i = i + 1
    }

    cout << " The sum of numbers from 1 to " <<  n << " is "
            << sum << endl;
}
```

Increment and decrement operators

	value	side effect
++x	x + 1	x = x + 1
x++	x	x = x + 1
--x	x - 1	x = x - 1
x--	x	x = x - 1

Examples:

```
x = 3;
a = ++x;    // x = 4, a = 4
b = --a;    // a = 3, b = 3
c = b++;    // c = 3, b = 4
d = c--;    // d = 3, c = 2
```

While Loop (Example)

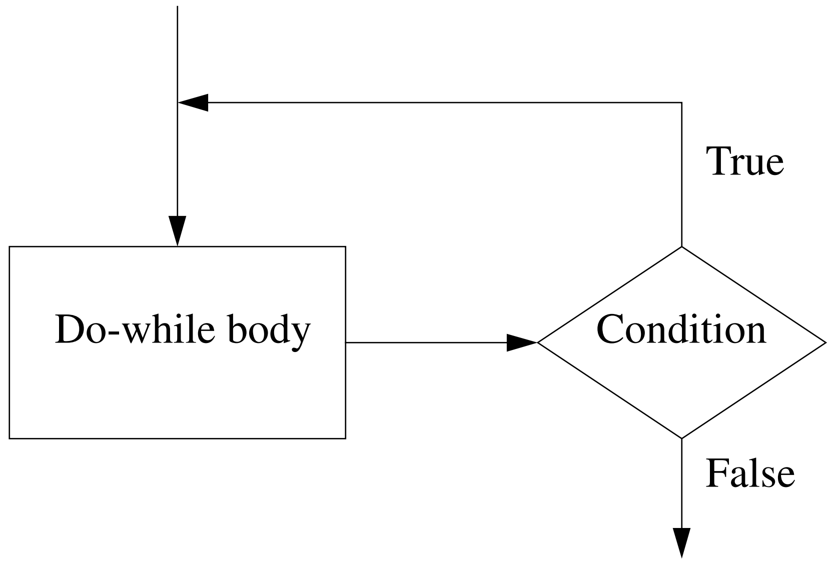
```
// Program to raise a number to the nth power
// Input:      number and exponent "n"
// Output:     number to the nth power

#include <iostream.h>

void main ( ) {
    int  number,  n,        // Input number & exponent
         result = 1;        //  Initialize result to 1

    cout << "Enter number: ";    cin >> number;
    cout << "Enter exponent: ";  cin >> n;

    while (n  >=  1) {
        result = result * number;
        n--;                           //  n = n -1
    }

    cout << number  << " to the power " <<  n << " = "
          << result << endl;
}
```

DO-WHILE Loop

do {

< sequence of statements >

} while (< logical expression >);



1. The statements will be executed at least once.

2. After all the statements have been executed, the logical expression will be evaluated. If the value is TRUE, then the control will go back to the beginning of the do-while loop. Otherwise, the loop terminates.

Do-While (Example)

```
// Find the maximum value of a list of position numbers
// Input:    A list of numbers
// Output:  Maximum value of the numbers entered

void main( )
{
int  value,                               //Input value
     maxval;                              //Maximum value

     //  value = 0;                       // Initialize value
     maxval = 0;                          // Initialize maxval

     do  {
          cout << "Enter a value (-1 to stop): ";
          cin >> value;

          if (value > maxval)
             maxval = value;              // new maximum
     }  while  (value != -1) ;

     cout << "The maximum value found is "
         << maxval << endl;
}
```

Do-While (Example)

```
//  Control loop for menu-driven program that prints a list
//  choices from which the program user selects a desired
//  operation.

 do {
   //  Display the menu
   cout << "1)  Compute an average. " << endl
        << "2)  Compute a standard deviation. " << endl
        << "3)  Find the median. " << endl
        << "4)  Find the smallest value. " << endl
        << "5)  Find the largest value. " << endl
        << "6)  Exit. " << endl;

   //  Read the user's choice
   cout << "Enter a number between 1 and 6: " << endl;
   cin >> choice;

   // Perform the user's choice
   do_choice (choice);

 }   while ( choice != 6) ;
```

For Statement

- Syntax

```
for (<initialize expression>;
         <test expression>;
         <update expression>) {
    <sequence of statements>
}
```

- Meaning

```
initialize loop control variable (lcv)
while test expression involving lcv is true
begin
  perform sequence of statements
  update lcv to next value
end
```

- Example

```
for (i=1; i<10; i++) {
  cout << i;
}

is the same as:

i = 1;
while ( i < 10) {
    cout << i;
    i++;
}

Both produce the output:
123456789
```

For Loop (Example)

```
// Program to raise a number to the nth power
// Input:     number and exponent "n"
// Output:    number to the nth power

#include <iostream.h>

void main ( ) {
    int  number,  n,        // Input number & exponent
         result = 1;        //  Initialize result to 1

    cout << “Enter number: “;    cin >> number;
    cout << “Enter exponent: “;  cin >> n;

    for  (int i = n;  i > 0;  i--) {
        result = result * number;
    }

    cout << number  << " to the power " <<  n << " = "
          << result << endl;
}
```

Nested Loops

- Example: Print out the following diamond pattern

```
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

- Subproblem:
  - o Print out the upper half
  - o Print out the lower half

- Print out the upper half:

```
row 1: print 4 space, 1 asterisk;
row 2: print 3 space, 3 asterisk;
row 3: print 2 space, 5 asterisk;
row 4: print 1 space, 7 asterisk;
row 5: print 0 space, 9 asterisk;
```

***Refinement*:**

```
row 1: print (5-row)space,(2*row-1)asterisk;
row 2: print (5-row)space,(2*row-1)asterisk;
row 3: print (5-row)space,(2*row-1)asterisk;
row 4: print (5-row)space,(2*row-1)asterisk;
row 5: print (5-row)space,(2*row-1)asterisk;
```

Nested Loops (Cont'd)

```
// Print out a diamond pattern
#include <iostream.h>

void main() {
   int row, space, asterisk;

   // top half
   for (row = 1; row <= 5; row++) {

      for (space = 1; space <= 5 - row; space++)
         cout << ' ';

      for (asterisk = 1;
           asterisk <= 2 * row - 1; asterick++)
         cout << '*';

      cout << endl;
   }

   // bottom half
   for (row = 4; row >= 1; row--) {

      for (space = 1; space <= 5 - row; space++)
         cout << ' ';

      for (asterisk = 1;
           asterisk <= 2 * row - 1; asterick++)
         cout << '*';

      cout << endl;
   }

}
```