## Control Structures

**Flow of control**

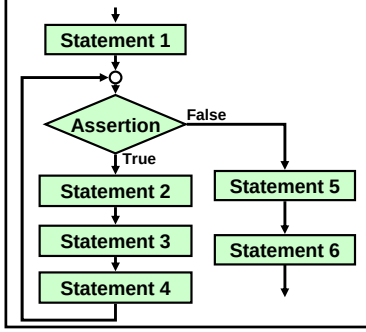
Execution sequence of program statements

- Sequential Control Structure
- Selection (Branching) Control Structure
- Repetition (Loop) Control Structure
  - Conditional T/F Assertion
  - Relational & Logical Operators for Assertion
  - Repetition processing
  - while( )  do-while( )  for( )

 1

## Repetition (Loop) Structure

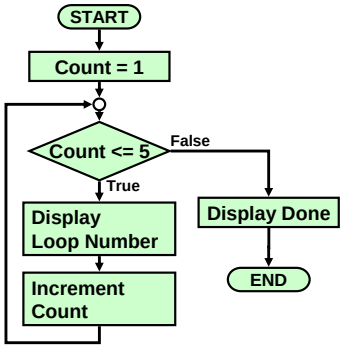
- Control structure used to repeat a sequence of instructions in a loop
- The simplest loop structure is the while()




```
Statement 1;
while(Assertion)
{
      Statement 2;
      Statement 3;
      Statement 4;
}
Statement 5;
Statement 6;
```

 2

## Repetition (Loop) Structure

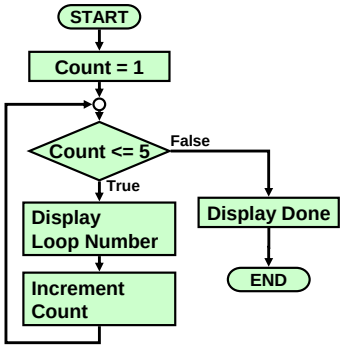



*Pseudo code:*

```
START
   Count = 1
   WHILE Count <= 5
      Display: Count
      Count = Count +1
   ENDWHILE
   Display: Done
END
```

 3

## Repetition (Loop) Structure




```
#include <iostream>
using namespace std;
int main()
{
  int nCnt = 1;
  while(nCnt <= 5)
  {
     cout << "In Loop "
     << nCnt << endl;
     nCnt = nCnt + 1;
  }
  cout << "Done";
  return 0;
}
```

 4

## while statement loop control

- Contents of loop executed repeatedly while(*assertion)* is true
- Loop terminated when while(assertion) is false.
- Counter-Controlled Loop Structure
  - Initialize a counter to count loops
  - Increment or decrement counter within loop
  - while(assertion) Counter value valid?
- Sentinel-Controlled Loop Structure
  - while(assertion) checks for a sentinel termination value

 5

## Counter-Controlled Repetition Structure

```
1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.    int nScore, nTotal=0, nCntr=0;
6.    while(nCntr < 5)
7.    {
8.      cout << "\nEnter Score\n>";
9.      cin >> nScore;
10.     nTotal = nTotal + nScore;
11.     nCntr = nCntr + 1;
12.     cout << "Score " << nCntr
13.          << " = " << nScore << endl;
14.   }
15.   cout << "----------------------"
16.        << "\nThe Average Score = "
17.        << nTotal/5 << endl;
18.  return 0;
19. }
```

```
Enter Score
>90
Score 1 = 90

Enter Score
>100
Score 2 = 100

Enter Score
>80
Score 3 = 80

Enter Score
>60
Score 4 = 60

Enter Score
>70
Score 5 = 70
----------------------
The Average Score = 80
```

 6

## Sentinel-Controlled Repetition Structure

```
1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.    int nScore=0, nTotal=0, nCntr=0;
6.    while(nScore >= 0)
7.    {
8.      cout << "\nEnter Score\n>";
9.      cin >> nScore;
10.     if(nScore >= 0)
11.     {
12.       nTotal = nTotal + nScore;
13.       cout << "Score " << ++nCntr
14.            << " = " << nScore << endl;
15.     }
16.   }
17.   cout << "----------------------"
18.        << "\nThe Average Score = "
19.        << nTotal/nCntr << endl;
20.  return 0;
21. }
```

```
Enter Score
>90
Score 1 = 90

Enter Score
>80
Score 2 = 80

Enter Score
>83
Score 3 = 83

Enter Score
>87
Score 4 = 87

Enter Score
>-1
----------------------
The Average Score = 85
```

 7

## Repetition Practice Programs

- Create a program that will display the first 10 multiples of an entered number.
  - Will loop be counter or sentinel controlled?
  - Example: if 7 is entered the format is:

    1 x 7 = 7

    2 x 7 = 14

    3 x 7 = 21 ...
- Create a program that will add scores until a -1 is entered
  - Will loop be counter or sentinel controlled?

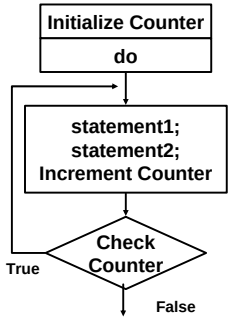
 8

## Filtered Input Application: *Wrong Entry is not an option!*

```
1.  #include <iostream>
2.  using namespace std;
3.  int main( )
4.  {
5.    char cEntry = 'f';
6.    while(cEntry != 'y' && cEntry != 'n')
7.    {
8.      cout << "Do you like Programming? (y or n)\n>";
9.      cin >> cEntry;
10.     cin.ignore(100,'\n');

11.     if(cEntry == 'y')
12.       cout << "I\'m glad you like programming!";
13.     else if(cEntry == 'n')
14.       cout << "You will like it if you study.";
15.     else
16.       cout << "You must enter either y or n!\n\n";
17.   }

18.   cout << "\n\nDone\n\n";
19.   return 0;
20. }
```

## do - while Structure

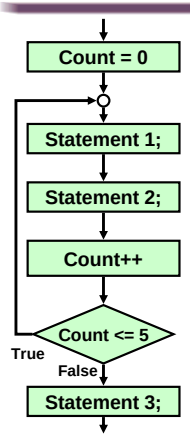
- A loop structure that guarantees the loop body is executed once.
- Condition is tested at bottom of loop




```
Initialize Counter;
REPEAT
      Statement1
      Statement2
      Increment Counter
UNTIL(Assertion)
```

 10

## do-while Example




- Don't forget semicolon for while(...);

```
1.  int nCount=0;
2.  do
3.  {
4.       Statement 1;
5.       Statement 2;
6.       nCount++;
7.  }while(nCount <=0);
8.  Statement 3;
```

Required

 11

## Filtered Input Application: *Using do – while*

Loop is guarenteed to execute one time because assertion checked at end of loop

```
#include <iostream>
using namespace std;
int main( )
{
  char cEntry; // No initialization necessary
  do
  {
    cout << "Do you like Programming? (y or n)\n>";
    cin >> cEntry;
    cin.ignore(100,'\n');
    if(cEntry == 'y')
      cout << "I\'m glad you like programming!";
    else if(cEntry == 'n')
      cout << "You will like it if you study.";
    else
      cout << "You must enter either y or n!\n\n";
  }while(cEntry != 'y' && cEntry != 'n');
  cout << "\n\nDone\n\n";
  return 0;
}
```

## Designing Loops

1. What condition will end the loop?
2. How should condition be initialized?
3. How should condition be updated?
4. What is the process being repeated?
5. How should the process be initialized?
6. How should the process be updated?
7. What is the state of the program on exiting the loop?

 13

## Temperature Program -Ver.3

```
1.  #include <iostream>
2.  #include <iomanip>
3.  using namespace std;
4.  int main()
5.  {
6.    // DECLARATION SECTION
7.    char cQuestion;
8.    float fTemperature;
9.    cout << fixed;   // Allows float point format

10.   // PROCESSING SECTION
11.   cout << "This program converts temperatures between\n"
12.        << "degrees Celsius and degrees Fahrenheit.\n"
13.        << "You may enter either a Celsius or "
14.        << "Fahrenheit\ntemperature for conversion.\n\n";
15.   while(true)
16.   {
17.     cout << "> <-- Enter C (Celsius), F (Fahrenheit), or Q (Quit)\r>";
18.     cin >> cQuestion;
19.     cin.ignore(100,'\n');
```

 14

```
20.     if(cQuestion == 'C' || cQuestion == 'c')
21.     {
22.       cout << "      <-- Enter temperature in degrees Celsius\r>";
23.       cin >> fTemperature;
24.       cin.ignore(100,'\n');
25.       cout << "Results: " << setprecision(2)
26.         << fTemperature << " C = "
27.         << (((fTemperature * 180) / 100) + 32) << " F\n";
28.     }
29.     else if(cQuestion == 'F' || cQuestion == 'f')
30.     {
31.       cout << "      <-- Enter temperature in degrees Fahrenheit  \r>";
32.       cin >> fTemperature;
33.       cin.ignore(100,'\n');
34.       cout << "Results: " << setprecision(2)
35.          << fTemperature << " F = "
36.          << (((fTemperature - 32) * 100) / 180) << " C\n";
37.     }
38.     else if (cQuestion == 'Q' || cQuestion == 'q')
39.     {
40.       cout << "\nGood bye";
41.       return 0;
42.     }
43.     else
44.       cout << "\a";
45.   }
46. }
```

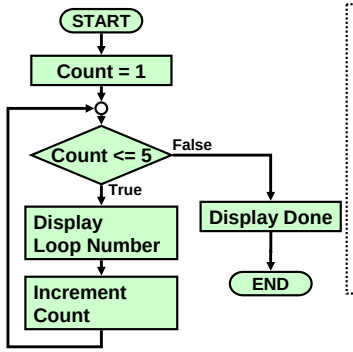
## Temperature Program Output

```
This program converts temperatures between
degrees Celsius and degrees Fahrenheit.
You may enter either a Celsius or Fahrenheit
temperature for conversion.

>r<-- Enter C (Celsius), F (Fahrenheit), or Q (Quit)
>W<-- Enter C (Celsius), F (Fahrenheit), or Q (Quit)
>C<-- Enter C (Celsius), F (Fahrenheit), or Q (Quit)
>100   <-- Enter temperature in degrees Celsius
Results: 100.00 C = 212.00 F
>f<-- Enter C (Celsius), F (Fahrenheit), or Q (Quit)
>0     <-- Enter temperature in degrees Fahrenheit
Results: 0.00 F = -17.78 C
>q<-- Enter C (Celsius), F (Fahrenheit), or Q (Quit)

Good bye

Terminated with return code 0
Press any key to continue ...
```

 16

## for Loop Structure

- Loop structure that contains in one statement:
  for( initialization; assertion; increment )




```
using namespace std;
int main()
{
  int nC = 1;
  for(nC=1; nC<= 5; nC++)
  {
    cout << "In Loop "
       << nC << endl;
  }
  cout << "Done";
  return 0;
}
```

 17

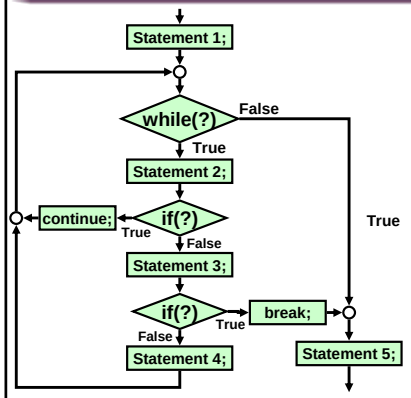
## Nested for Loop Example

```
#include <iostream>
using namespace std;
int main(void)
{
  int nI, nJ;
  for(nI=1; nI<=5; nI++)
    cout << nI << " ";
  cout << endl;
  for(nI = 5; nI >= 0; --nI)
  {
    cout << endl;
    for( nJ = 4; nJ >= 0; --nJ)
    {
      cout << (nI + nJ) << ' ';
    }
  }
  return 0;
}
```

```
1 2 3 4 5

9 8 7 6 5
8 7 6 5 4
7 6 5 4 3
6 5 4 3 2
5 4 3 2 1
4 3 2 1 0
```

 18

## break; continue; commands




```
1. Statement 1;
2. while(?)
3. {
4.   Statement 2;
5.   if(?)
6.     continue;
7.   Statement 3;
8.   if(?)
9.     break;
10.  Statement 4;
11.}
12.Statement 5;
```

 19

```
1.  #include <iostream>
2.  using namespace std;
3.  int main( )
4.  {
5.    char cEntry = 'f';
6.    while(true)
7.    {
8.      cout << "Do you like Programming? (y or n)\n>";
9.      cin >> cEntry;
10.     cin.ignore(100,'\n');

11.     if(cEntry == 'y')
12.     {
13.       cout << "I\'m glad you like programming!";
14.       break;
15.     }
16.     if(cEntry == 'n')
17.     {
18.       cout << "You will like it if you study.";
19.       break;
20.     }
21.     cout << "You must enter either y or n!\n\n";
22.   }
23.   cout << "\n\nDone\n\n";
24.   return 0;
25. }
```

```
1.  #include <iostream>
2.  using namespace std;
3.  int main( )
4.  {
5.    int nEntry;
6.    cout << "This program will allow you to enter\n"
7.         << "numbers in the range 20 to 100\n\n";
8.    while(true)
9.    {
10.     cout << "Enter number: ";
11.     cin >> nEntry;
12.     cin.ignore(100,'\n');
13.     if(nEntry <= 100 && nEntry >= 20)
14.     {
15.       cout << "Entry " << nEntry << " is valid\n";
16.       break;
17.     }
18.     if(nEntry < 20)
19.     {
20.       cout << "Your entry is less then 20!\n\n";
21.       continue;
22.     }
23.     cout << "Your entry is greater then 100!\n\n";
24.   }
25.   cout << "\n\nDone\n\n";
26.   return 0;
27. }
```

## switch - case Structure

- A selection structure that can be used to select one of many branches
- Works best for a switch variable type of Integers or Characters but not strings

```
switch (Grade)
{
      case 'P': case 'p':
            cout << "You Passed\n";
            break;
      case 'F': case 'f':
            cout << "You Failed";
            break;
      default:
            cout <<"Incorrect letter Grade.";
}
```

 22

```
1.  #include <iostream>
2.  using namespace std;
3.  int main( )
4.  {
5.    char cEntry;
6.    cout << "This program will make a"
7.         << "comment about your grade.\n\n";
8.    cout << "Enter Grade: ";
9.    cin >> cEntry;
10.   cin.ignore(100,'\n');
11.   switch (cEntry)
12.   {
13.     case 'A': case 'a':
14.       cout << "Excellent work";
15.       break;
16.     case 'B': case 'b':
17.       cout << "Good work";
18.       break;
19.     case 'C': case 'c':
20.       cout << "Average work";
21.       break;
22.     case 'D': case 'd':
23.     case 'F': case 'f':
24.       cout << "Poor work";
25.       break;
26.     default: // catch all other characters
27.       cout << "Incorrect letter Grade.";
28.   }
29.   cout << "\n\nDone\n\n";
30.   return 0;
31. }
```