**University of Management and Technology, Lahore Campus**

**Lab- 07 Manual**

Lab Instructor: Riaz Ahmad

Department of Computer Science

Email: [riazahmad@umt.edu.pk](mailto:riazahmad@umt.edu.pk)

**Increment /Decrement operator & For Loop & Switch Statement**

**Coding Examples:**

**Decrement operator:**

include <iostream>

using namespace std;

int main()

{

int x=10,a;

a=--x;

cout<<"pre decrement operator";

cout<<"\na = "<<a;

cout<<"\nx ="<<x;

return 0;

}

**Output:**

Pre decrement operator

a=9

x=9

include <iostream>

using namespace std;

int main()

{

int x=10,a;

a=x--;

cout<<"Post decrement operator";

cout<<"\na = "<<a;

cout<<"\nx ="<<x;

return 0;

}

**Output:**

a=10

x=9

**Increment Operator:**

include <iostream>

using namespace std;

int main()

{

int x=10,a;

a=++x;

cout<<"pre increment operator";

cout<<"\na = "<<a;

cout<<"\nx ="<<x;

return 0;

}

**Output:**

Pre increment operator

a=11

x=11

include <iostream>

using namespace std;

int main()

{

int x=10,a;

a=x++;

cout<<"Post increment operator";

cout<<"\na = "<<a;

cout<<"\nx ="<<x;

return 0;

}

**Output:**

Post increment operator

a=10

x=11

**For Loop:**

* Loop: a control structure that causes a statement or statements to repeat
* Useful for counter-controlled loop
* General Format:

for(initialization; test; update)

statement; // or block in { }

* No semicolon after the update expression or after the )

**for Loop - Mechanics**

for(*initialization*; *test*; *update*)

*statement*; // or block in { }

1. Perform *initialization*
2. Evaluate *test* expression
   * If true, execute *statement*
   * If false, terminate loop execution
3. Execute *update*, then re-evaluate *test* expression

**Example:**

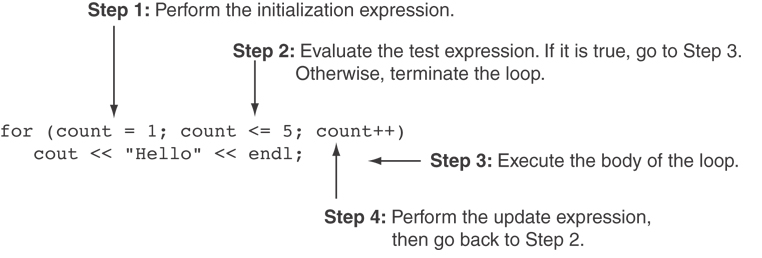
**Code:**

int count;

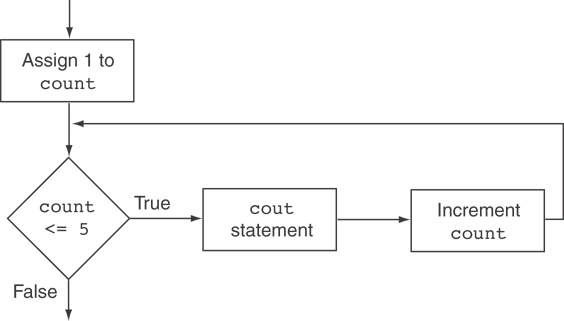
for (count = 1; count <= 5; count++)

cout << "Hello" << endl;

**Execution Process:**

****

**Flow Chart:**

****

**Example 1: Write a program in C++ to find the first 10 natural numbers.**

#include <iostream>

using namespace std;

int main()

{

int i;

cout << "\n\n Find the first 10 natural numbers:\n";

cout << "---------------------------------------\n";

cout << " The natural numbers are: \n";

for (i = 1; i <= 10; i++)

{

cout << i << " ";

}

cout << endl;

return 0;

}

**Example 2: Write a program in C++ to find the sum of first 10 natural numbers.**

#include <iostream>

using namespace std;

int main()

{

int i,sum=0;

cout << "\n\n Find the first 10 natural numbers:\n";

cout << "---------------------------------------\n";

cout << " The natural numbers are: \n";

for (i = 1; i <= 10; i++)

{

cout << i << " ";

sum=sum+i;

}

cout << "\n The sum of first 10 natural numbers: "<<sum << endl;

}

**Example 3: Write a program in C++ to display n terms of natural number and their sum.**

#include <iostream>

using namespace std;

int main()

{

int n,i,sum=0;

cout << "\n\n Display n terms of natural number and their sum:\n";

cout << "---------------------------------------\n";

cout << " Input a number of terms: ";

cin>> n;

cout << " The natural numbers upto "<< n <<"the terms are: \n";

for (i = 1; i <= n; i++)

{

cout << i << " ";

sum=sum+i;

}

cout << "\n The sum of the natural numbers is: "<<sum << endl;

}

**Example 4: Write a program in C++ to find the factorial of a number.**

#include <iostream>

using namespace std;

int main()

{

int num1,factorial=1;

cout << "\n\n Find the factorial of a number:\n";

cout << "------------------------------------\n";

cout << " Input a number to find the factorial: ";

cin>> num1;

for(int a=1;a<=num1;a++)

{

factorial=factorial\*a;

}

cout<<" The factorial of the given number is: "<<factorial<<endl;

return 0;

}

**Example 5: A person invests $ 1000.00 in a savings account yielding 5 % interest. Assuming that all interests is left on deposit in the account, calculate and print the amount of money in the account at the end of each year for ten years. Use the following formula for determining these amounts:**

**a = p(1 + r) ^2**

where

p = original amount invested (Principal),

r = annual interest rate,

n = number of years, and

a = amount on deposit at the end of the nth year.

//Calculation of Compound Interest

#include <iostream>

#include <iomanip>

#include <math.h>

using namespace std;

int main()

{

double amount, // amount on deposit

principal = 1000.0, // starting principal

rate = .05; // interest rate

cout << "Year" << setw( 21 )<< "Amount on deposit" << endl;

cout << "--------------------------------------------\n";

for ( int year = 1; year <=10; year++ )

{

amount = principal \* pow( 1.0 + rate, year );

cout << setw( 4 ) << year <<fixed<< setprecision( 2 )

<< setw( 21 )<< amount << endl;

}

return 0;

}

**Example 6: Write a program in C++ to check whether a number is prime or not.**

#include <iostream>

using namespace std;

int main()

{

int num1, ctr = 0;

cout << "\n\n Check whether a number is prime or not:\n";

cout << "--------------------------------------------\n";

cout << " Input a number to check prime or not: ";

cin>> num1;

for (int a = 1; a <= num1; a++)

{

if (num1 % a == 0)

{

ctr++;

}

}

if (ctr == 2)

{

cout << " The entered number is a prime number. \n";

}

else {

cout << " The number you entered is not a prime number. \n";

}

return 0;

}

**// Using the Break statement in a for structure**

#include <iostream>

using namespace std;

int main()

{

// x declared here so it can be used after the loop.

int x;

for ( x = 1; x <= 10; x++ )

{

if ( x == 5 )

break; // break loop only if x is 5

cout << x << " ";

}

cout << "\nBroke out of loop at x of " << x << endl;

return 0;

}

**// Using the continue statement in a for structure**

#include <iostream>

using namespace std;

int main()

{

for ( int x = 1; x <= 10; x++ )

{

if ( x == 5 )

continue; // skip remaining code in loop only if x is 5

cout << x << " ";

}

cout << "\nUsed continue to skip printing the value 5" << endl;

return 0;

}

***THE switch MULTIPLE-SELECTION STRUCTURE***

**Example 7: Basic Calculator using Switch statement.**

# include <iostream>

using namespace std;

int main()

{

char op;

float num1, num2;

cout << "Enter operator: +, -, \*, /: ";

cin >> op;

cout << "Enter two operands: ";

cin >> num1 >> num2;

switch(op)

{

**case '+':**

cout << num1 << " + " << num2 << " = " << num1 + num2;

break;

**case '-':**

cout << num1 << " - " << num2 << " = " << num1 - num2;

break;

**case '\*':**

cout << num1 << " \* " << num2 << " = " << num1 \* num2;

break;

**case '/':**

cout << num1 << " / " << num2 << " = " << num1 / num2;

break;

**default:**

// If the operator is other than +, -, \* or /, error message is shown

cout << "Error! operator is not correct";

break;

}

return 0;

}

**Example 8: Menu based Basic Calculator using Switch statement.**

/\* C++ program to implement simple functions of a calculator\*/

#include<iostream>

using namespace std;

int main()

{

int choice;

long num1, num2, x;

//displaying different options

cout << "Please choose your option:"

"\n1 = Addition"

"\n2 = Subtraction"

"\n3 = Multiplication"

"\n4 = Division"

"\n5 = Squares"

"\n6 = exit"

"\n\nChoice: ";

cin >> choice;

//while loop check whether the choice is in the given range

while(choice < 1 || choice > 6)

{

cout << "\nPlease choose the above mentioned option."

"\nChoice: ";

cin >> choice;

}

switch (choice)

{

//Addition

case 1:

cout << "Enter two numbers: \n";

cin >> num1 >> num2;

x = num1 + num2;

cout << "Sum = " << x;

break;

//Subtraction

case 2:

cout << "Enter two numbers: \n";

cin >> num1 >> num2;

x = num1 - num2;

cout << "Subtraction = " << x;

break;

//Multiplication

case 3:

cout << "Enter two numbers: \n";

cin >> num1 >> num2;

x = num1 \* num2;

cout << "Product = " << x;

break;

//Division

case 4:

cout << "Enter Dividend: ";

cin >> num1;

cout << "Enter Divisor: ";

cin >> num2;

//while loop checks for divisor whether it is zero or not

while(num2 == 0)

{

cout << "\nDivisor cannot be zero."

"\nEnter divisor once again: ";

cin >> num2;

}

x = num1 / num2;

cout << "\nQuotient = " << x;

break;

//Square

case 5:

cout << "Enter any number: \n";

cin >> num1;

x = num1 \* num1;

cout << "Square = " << x;

break;

case 6:

return 0;

default: cout << "\nError";

}

}

**Example 9: Count grades using Switch statement.**

# include <iostream>

using namespace std;

int main()

{

int grade, // one grade

aCount = 0, // number of A's

bCount = 0, // number of B's

cCount = 0, // number of C's

dCount = 0, // number of D's

fCount = 0; // number of F's

cout << "Enter the letter grades." << endl

<< "Enter the EOF character to end input." << endl;

while ( ( grade = cin.get() ) !='1' ) {

// EOF is system dependent. Try CTRL-Z

switch ( grade ) { // switch nested in while

// case and default need a colon :,

// not a semicolon.

case 'A': // grade was uppercase 'A'

case 'a': // or lowercase 'a'

++aCount;

break; // necessary to exit switch

case 'B': // grade was uppercase 'B'

case 'b': // or lowercase 'b'

++bCount;

break;

case 'C': // grade was uppercase 'C'

case 'c': // or lowercase 'c'

++cCount;

break;

case 'D': // grade was uppercase 'D'

case 'd': // or lowercase 'd'

++dCount;

break;

case 'F': // grade was uppercase 'F'

case 'f': // or lowercase 'f'

++fCount;

break;

// for additional protection

case '\n': // ignore newlines,

case '\t': // tabs,

case ' ': // and spaces in input

break;

default: // catch all other characters

cout << "Incorrect letter grade entered."

<< "Enter a new grade." << endl;

break; // optional

}

}

cout << "\n\nTotals for each letter grade are:"

<< "\nA: " << aCount

<< "\nB: " << bCount

<< "\nC: " << cCount

<< "\nD: " << dCount

<< "\nF: " << fCount << endl;

return 0;

}