## Programming Fundamentals

## LAB MANUAL 9

## 

**Implementation of different programs using break, continue and different library functions**

**C++ break Statement**

The break; statement terminates the loop (for, while and do..while loop) and switch statement immediately when it appears.

**1- C++ program to add all number entered by user until user enters 0.**

// C++ Program to demonstrate working of break statement

#include <iostream>

using namespace std;

int main() {

float number, sum = 0.0;

while (true) { // test expression is always true

cout<<"Enter a number: ";

cin>>number;

if (number != 0.0) {

sum += number;

}

else {

break; // terminating the loop if number equals to 0.0

}

}

cout<<"Sum = "<<sum;

return 0;

}

02—

// **break can be used to get out of an infinite loop:**

#include <iostream>

using namespace std;

int main()

{

int num=1;

while (num!=0) // infinite loop

{

cout << "Enter 0 to exit or anything else to continue: ";

cin >> num;

// exit loop if user enters 0

if (num == 0)

break;

}

cout << "We're out!\n";

return 0;

}

**Continue Statement:**

The continue statement provides a useful way to jump back to the top of a loop earlier than normal, which can be used to bypass the remainder of the loop for an iteration. Causes the remaining portion of the enclosing [for](http://en.cppreference.com/w/cpp/language/for),  [while](http://en.cppreference.com/w/cpp/language/while) or [do-while](http://en.cppreference.com/w/cpp/language/do) loop body to be skipped.

*2****-* //This program is intended to print every number between 0 and 9 except 5. But it actually //prints:**

**//0 1 2 3 4**

#include <iostream>

using namespace std;

int main() {

int count(0);

while (count < 10)

{

    if (count == 5)

        continue;

    cout << count << " ";

    ++count;

}

}

**3-**

#include <iostream>

using namespace std;

int main ()

{

// Local variable declaration:

int a = 10;

// do loop execution

do

{

if( a == 15)

{

// skip the iteration.

a = a + 1;

continue;

}

cout << "value of a: " << a << endl;

a = a + 1;

}while( a < 20 );

return 0;

}

**4 –**

#include <iostream>

using namespace std;

int main()

{

for (int i = 0; i < 10; i++)

{

if (i != 5) continue;

cout << i << " "; //this statement is skipped each time i!=5

}

cout << '\n';

cout << '\n';

for (int j = 0; j < 2; j++)

{

for (int k = 0; k < 5; k++)

{ //only this loop is affected by continue

if (k == 3) continue;

cout << j << k << " "; cout << '\n'; //this statement is skipped each time k==3

}

}

return 0;

}

**Library function exit ():**

This function causes the program to terminate, no matter where it is in the listing. It has no return value. Its single argument, 0 in our next example, is returned to the operating system when the program exits.

|  |
| --- |
| **--5**  #include<iostream>  #include<conio.h>  using namespace std;  main()  {  int n,j;  cout<< "Enter a number: ";  cin>> n; //get number to test  for(j=2; j <= n/2; j++) //divide by every integer from  {  if(n%j == 0) //2 on up; if remainder is 0,  {  // it’s divisible by j  cout<< "It’s not prime; divisible by " << j <<endl;  exit(0); //exit from the program  } // if ends here  } // for ends here  cout<< "It's prime\n";  getch();  } |

**getche() library function():**

So far we’ve used only cin and >> for input. That approach requires that the user always press the Enter key to inform the program that the input is complete. This is true even for single characters: The user must type the character, and then press Enter. However, as in the next example, program needs to process each character typed by the user without waiting for an Enter.

**The getche() library function performs this service. It returns each character as soon**

**as it’s typed.**

|  |
| --- |
| **--06**  #include <iostream>  using namespace std;  #include <conio.h> // for getche()  main()  {  int chcount=0;  char ch;  cout<<"Enter phrase: ";  while( (ch=getche()) != '\r' ) // loop until Enter typed  {  chcount++; // count a character  } // display results  cout<<endl<<endl <<"Letters : "<<chcount <<endl;  getch();  } |

**Task 01:**

Modify the above program so that user enters a phrase and your program calculates the number of words as well as number of characters in a phrase.

Output can be like this:

|  |
| --- |
| Enter a phrase: Hello world  Words: 2  Letters: 10 |

**Task 02:**

Write a C++ Program that allows the user to type up to 10 numbers, and displays the sum of all the numbers entered at the end. If the user enters 0, the break causes the loop to terminate early (before 10 numbers have been entered).

**Task 03:**

Create the equivalent of a four-function calculator. The program should ask the user to enter a number, an operator, and another number. (Use floating point.) It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. Use a switch statement to select the operation. Finally, display the result.

When it finishes the calculation, the program should ask whether the user wants to do another calculation. The response can be ‘y’ or ‘n’. Some sample interaction with the program might look like this:

|  |
| --- |
| Enter first number, operator, second number: 10 / 3  Answer = 3.333333  Do another (y/n)? y  Enter first number, operator, second number: 12 + 100  Answer = 112  Do another (y/n)? n |