```
// Program to build a simple calculator using switch Statement
#include <iostream>
using namespace std;

int main() {
    char oper;
    float num1, num2;
    cout << "Enter an operator (+, -, *, /): ";
    cin >> oper;
    cout << "Enter two numbers: " << endl;
    cin >> num1 >> num2;

    switch (oper) {
        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:
         // operator is doesn't match any case constant (+, -, *, /)
         cout << "Error! The operator is not correct";
         break;
   }

   return 0;
}
```

```
// Fig. 2.17: fig02_17.cpp
// Counter-controlled repetition with the for structure.
#include <iostream>

using std::cout;
using std::endl;

// function main begins program execution
int main()
{
  // Initialization, repetition condition and incrementing
  // are all included in the for structure header.

  for ( int counter = 1; counter <= 10; counter++ )
    cout << counter << endl;

  return 0;   // indicate successful termination

} // end function main
```

```
// Fig. 2.20: fig02_20.cpp
// Summation with for.
#include <iostream>

using std::cout;
using std::endl;

// function main begins program execution
int main()
{
  int sum = 0;                    // initialize sum

  // sum even integers from 2 through 100
  for ( int number = 2; number <= 100; number += 2 )
    sum += number;              // add number to sum

  cout << "Sum is " << sum << endl;  // output sum
  return 0;                     // successful termination

} // end function main
```

```
// Fig. 2.21: fig02_21.cpp
// Calculating compound interest.
#include <iostream>

using std::cout;
using std::endl;
using std::ios;
using std::fixed;

#include <iomanip>

using std::setw;
using std::setprecision;
```

```
#include <cmath>  // enables program to use function pow

// function main begins program execution
int main()
{
   double amount;             // amount on deposit
   double principal = 1000.0;  // starting principal
   double rate = .05;         // interest rate

   // output table column heads
   cout << "Year" << setw( 21 ) << "Amount on deposit" << endl;

   // set floating-point number format
   cout << fixed << setprecision( 2 );

   // calculate amount on deposit for each of ten years
   for ( int year = 1; year <= 10; year++ ) {

      // calculate new amount for specified year
      amount = principal * pow( 1.0 + rate, year );

      // output one table row
      cout << setw( 4 ) << year
           << setw( 21 ) << amount << endl;

   } // end for

   return 0;   // indicate successful termination

} // end function main
```

```
// Fig. 2.22: fig02_22.cpp
// Counting letter grades.
#include <iostream>

using std::cout;
using std::cin;
using std::endl;

// function main begins program execution
int main()
{
   int grade;       // one grade
   int aCount = 0;  // number of As
   int bCount = 0;  // number of Bs
   int cCount = 0;  // number of Cs

   int dCount = 0;  // number of Ds

   int fCount = 0;  // number of Fs


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

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


   // loop until user types end-of-file key sequence

   while ( ( grade = cin.get() ) != EOF ) {


      // determine which grade was input

      switch ( grade ) {  // switch structure nested in while


         case 'A':        // grade was uppercase A

         case 'a':        // or lowercase a

            ++aCount;     // increment aCount

            break;        // necessary to exit switch


         case 'B':        // grade was uppercase B

         case 'b':        // or lowercase b

            ++bCount;     // increment bCount

            break;        // exit switch
```

```
case 'C':       // grade was uppercase C
case 'c':       // or lowercase c
  ++cCount;     // increment cCount
  break;        // exit switch

case 'D':       // grade was uppercase D
case 'd':       // or lowercase d
  ++dCount;     // increment dCount
  break;        // exit switch

case 'F':       // grade was uppercase F
case 'f':       // or lowercase f
  ++fCount;     // increment fCount
  break;        // exit switch

case '\n':      // ignore newlines,
case '\t':      // tabs,
case ' ':       // and spaces in input
  break;        // exit switch

default:        // catch all other characters
```

```
        cout << "Incorrect letter grade entered."
           << " Enter a new grade." << endl;
        break;       // optional; will exit switch anyway

    } // end switch

  } // end while

  // output summary of results
  cout << "\n\nTotals for each letter grade are:"
     << "\nA: " << aCount   // display number of A grades
     << "\nB: " << bCount   // display number of B grades
     << "\nC: " << cCount   // display number of C grades
     << "\nD: " << dCount   // display number of D grades
     << "\nF: " << fCount   // display number of F grades
     << endl;

  return 0;  // indicate successful termination

} // end function main
```

```
// Fig. 2.24: fig02_24.cpp
// Using the do/while repetition structure.
#include <iostream>

using std::cout;
using std::endl;

// function main begins program execution
int main()
{
  int counter = 1;            // initialize counter

  do {
    cout << counter << "  ";   // display counter
  } while ( ++counter <= 10 );  // end do/while

  cout << endl;

  return 0;   // indicate successful termination

} // end function main
```

```
// Fig. 2.26: fig02_26.cpp
// Using the break statement in a for structure.
#include <iostream>

using std::cout;
using std::endl;

// function main begins program execution
int main()
{

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

   // loop 10 times
   for ( x = 1; x <= 10; x++ ) {

      // if x is 5, terminate loop
      if ( x == 5 )
         break;          // break loop only if x is 5

      cout << x << " ";   // display value of x
   } // end for

   cout << "\nBroke out of loop when x became " << x << endl;

   return 0;   // indicate successful termination

} // end function main
```

```
// Fig. 2.27: fig02_27.cpp
// Using the continue statement in a for structure.
#include <iostream>

using std::cout;
using std::endl;

// function main begins program execution
int main()
{
   // loop 10 times
   for ( int x = 1; x <= 10; x++ ) {

      // if x is 5, continue with next iteration of loop
```

```
    if ( x == 5 )
      continue;       // skip remaining code in loop body

    cout << x << " ";   // display value of x

  } // end for structure

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

  return 0;            // indicate successful termination

} // end function main

#include <iostream>
using namespace std;

int main()
{
   cout << "\n\nWelcome to Studytonight :-)\n\n\n";
   cout << " =====  Program to find the GCD and LCM of two numbers ===== \n\n";
```

```
    //variable declaration
    int n1, n2, i;

    //variable declaration and initialization
    int gcd = 1, lcm = 1;

    //taking input from the command line (user)
    cout << " Enter the two numbers you want to find the GCD and LCM of : \n\n";
    cin >> n1 >> n2;

    //logic to calculate the GCD and LCM of the two numbers
    for ( i = 1; i < 10000; i++)
    {
        //i is the least value that perfectly divides both the numbers and hence the GCD
        if ((n1 % i == 0) && (n2 % i == 0))
        {
            gcd = i;
        }
    }
    lcm = (n1 * n2) / gcd;

    cout << " \n\nThe GCD of the two numbers : " << n1 << " and " << n2 << " is : " << gcd;
    cout << " \n\nThe LCM of the two numbers : " << n1 << " and " << n2 << " is : " << lcm << "\n\n";
    cout << "\n\n\n";

    return 0;
}
```

Greatest Common Divisor or GCD of two numbers is the maximum possible number that perfectly divides (remainder 0) both the numbers togother.

Example:

Consider two numbers to be 2 and 3. Now 12 has both 2 and 3 as its factors but 6 is the least possible number that has both 2 and 3 as its factors or 6 is the least number that is a multiple of both 2 and 3. Hence 6 is the LCM of 2 and 3.

What is a LCM?

Least Common Multiple or LCM of two numbers is the least possible number that is a multiple of both the numbers or that has both the numbers as its factors.

Example:

Consider two numbers to be 20 and 30. Now 1 perfectly divides both 20 and 30. Even 2 and 5 perfectly divides both 20 and 30. But 10 is the largest number that divides both 20 and 30 together and hence is considered to be the GCD of 20 and 30.