```
// Ex. 2.5: ex02_05.cpp
// Calculate the sum of the integers from 1 to 10.
#include <iostream>

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

// function main begins program execution
int main()
{
   int sum;              // stores sum of integers 1 to 10
   int x;              // counter

   x = 1;              // count from 1
   sum = 0;              // initialize sum

   while ( x <= 10 ) {
      sum += x;          // add x to sum
      ++x;              // increment x

   } // end while
   cout << "The sum is: " << sum << endl;

   return 0;  // indicate successful termination

} // end function main
```

```
// Ex. 2.8: ex02_08.cpp
// Raise x to the y power.
#include <iostream>

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

// function main begins program execution
int main()
{
   int x;      // base
   int y;      // exponent
   int i;     // counts from 1 to y
   int power;  // used to calculate x raised to power y
```

```
  i = 1;     // initialize i to begin counting from 1
  power = 1;  // initialize power

  cout << "Enter base as an integer: ";  // prompt for base
  cin >> x;                       // input base

  // prompt for exponent
  cout << "Enter exponent as an integer: ";
  cin >> y;                       // input exponent

  // count from 1 to y and multiply power by x each time
  while ( i <= y ) {
    power *= x;
    ++i;

  } // end while
  cout << power << endl;  // display result

  return 0;            // indicate successful termination

} // end function main
====================================================
//program prints pattern
#include <iostream>
using namespace std ;
int main ()
{
        int i , j ;
        for ( i=1 ; i<=3; i++){
                for ( j=1; j<=3;j++){
                        if ( i>= j)
                        cout <<"*";
                        else cout << " ";

                }
                cout<<endl;
        }
return 0;
}
```

```
//program prints pattern
#include <iostream>
```

```
using namespace std ;

int main ()

{

        int i , j ;

        int c = 1;

        for ( i=1 ; i<=3; i++){

                for ( j=1; j<=3;j++){

                        if ( i>= j){

                        cout << c;

                        c++;

                }

                        else cout << " ";


                }

                cout<<endl;

        }

return 0;

}
==================================================

// program prints pattern

#include <iostream>

using namespace std ;
```

```
int main ()
{
    int i , j ;
    int c = 0;
    for ( i=1 ; i<=3; i++){
        for ( j=1; j<=3;j++){
            if ( i>= j){

            cout << char (c + 'A');
            c++;
            cout << "  ";
        }
            else cout << "  ";

        }
        cout<<endl;
    }
return 0;
}
// program calculated factorial
#include <iostream>
using namespace std;
```

```
int main(){
int i=1,x,f;
cout<<"enter the number:";
cin>>x;
f=1;
while (i<=x)
{
f=f*i;
i++;
}
cout << f;
return 0;
}
```

```
// Fig. 2.7: fig02_07.cpp
// Class average program with counter-controlled repetition.
#include <iostream>

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

// function main begins program execution
```

```
int main()
{
  int total;       // sum of grades input by user
  int gradeCounter; // number of grade to be entered next
  int grade;       // grade value
  int average;     // average of grades

  // initialization phase
  total = 0;        // initialize total
  gradeCounter = 1;   // initialize loop counter

  // processing phase
  while ( gradeCounter <= 10 ) {      // loop 10 times
    cout << "Enter grade: ";         // prompt for input
    cin >> grade;                    // read grade from user
    total = total + grade;           // add grade to total
    gradeCounter = gradeCounter + 1;  // increment counter
  }

  // termination phase
  average = total / 10;              // integer division
```

```
   // display result
   cout << "Class average is " << average << endl;

   return 0;   // indicate program ended successfully

} // end function main
```

==

```
// Fig. 2.9: fig02_09.cpp
// Class average program with sentinel-controlled repetition.
#include <iostream>

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

#include <iomanip>        // parameterized stream manipulators

using std::setprecision;  // sets numeric output precision

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

```
{
  int total;       // sum of grades
  int gradeCounter;  // number of grades entered
  int grade;       // grade value

  double average;   // number with decimal point for average

  // initialization phase
  total = 0;       // initialize total
  gradeCounter = 0;  // initialize loop counter

  // processing phase
  // get first grade from user
  cout << "Enter grade, -1 to end: ";  // prompt for input
  cin >> grade;                  // read grade from user

  // loop until sentinel value read from user
  while ( grade != -1 ) {
    total = total + grade;          // add grade to total
    gradeCounter = gradeCounter + 1;  // increment counter

    cout << "Enter grade, -1 to end: ";  // prompt for input
    cin >> grade;                    // read next grade

  } // end while

  // termination phase
  // if user entered at least one grade ...
  if ( gradeCounter != 0 ) {

    // calculate average of all grades entered
    average = static_cast< double >( total ) / gradeCounter;

    // display average with two digits of precision
    cout << "Class average is " << setprecision( 2 )
        << fixed << average << endl;

  } // end if part of if/else

  else // if no grades were entered, output appropriate message
    cout << "No grades were entered" << endl;

  return 0;   // indicate program ended successfully
```

```
} // end function main
```

The fixed() method of stream manipulators in C++ is used to set the floatfield format flag for the specified str stream. This flag sets the floatfield to fixed. It means that the floating-point values will be written in fixed point notations.

```
// Fig. 2.11: fig02_11.cpp
// Analysis of examination results.
#include <iostream>

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

// function main begins program execution
int main()
{
   // initialize variables in declarations
   int passes = 0;          // number of passes
   int failures = 0;        // number of failures
   int studentCounter = 1;  // student counter
   int result;              // one exam result
```

```
// process 10 students using counter-controlled loop
while ( studentCounter <= 10 ) {

  // prompt user for input and obtain value from user
  cout << "Enter result (1 = pass, 2 = fail): ";
  cin >> result;

  // if result 1, increment passes; if/else nested in while
  if ( result == 1 )        // if/else nested in while
    passes = passes + 1;

  else  // if result not 1, increment failures
    failures = failures + 1;

  // increment studentCounter so loop eventually terminates
  studentCounter = studentCounter + 1;

} // end while

// termination phase; display number of passes and failures
cout << "Passed " << passes << endl;
```

```
  cout << "Failed " << failures << endl;

  // if more than eight students passed, print "raise tuition"
  if ( passes > 8 )
    cout << "Raise tuition " << endl;

  return 0;   // successful termination

} // end function main



// Fig. 2.14: fig02_14.cpp
// Preincrementing and postincrementing.
#include <iostream>

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

// function main begins program execution
int main()
{
  int c;                    // declare variable
```

```
   // demonstrate postincrement
   c = 5;                    // assign 5 to c
   cout << c << endl;        // print 5
   cout << c++ << endl;      // print 5 then postincrement
   cout << c << endl << endl; // print 6

   // demonstrate preincrement
   c = 5;                    // assign 5 to c
   cout << c << endl;        // print 5
   cout << ++c << endl;      // preincrement then print 6
   cout << c << endl;        // print 6

   return 0;   // indicate successful termination

} // end function main
=============================================
// Fig. 2.16: fig02_16.cpp
// Counter-controlled repetition.
#include <iostream>
using std::cout;
using std::endl;
```

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

   while ( counter <= 10 ) {    // repetition condition
      cout << counter << endl;  // display counter
      ++counter;               // increment
   } // end while

   return 0;                  // successful termination

} // end function main
//program prints pattern
#include <iostream>
using namespace std ;
int main ()
{
        int i , j ;
        for ( i=1 ; i<=3; i++){
                for ( j=1; j<=3;j++){
                    if ( i+ j >= 4)
                    cout <<"*";
                    else cout << " ";

                }
                cout<<endl;
        }
return 0;
}
```

```
#include <iostream>
using namespace std ;
int main ()
{
        int i,j ;
        for ( i=1 ; i<=9; i++){
                for ( j=1; j<=9;j++){
                if ((i+j>=6) && (j-i<=4) && (i-j<=4) && (i+j<=14))
```

```
                        cout <<"*";

                        else cout << " ";

                }
                cout<<endl;
        }
return 0;
}
```

```
#include <iostream>
using namespace std ;
int main ()
{
        int i,j ;
        for ( i=1 ; i<=9; i++){
                for ( j=1; j<=9;j++){
                if ((i+j==6) || (j-i==4) || (i-j==4) || (i+j==14))
```

```
                    cout <<"*";


                    else cout << " ";

                }
                cout<<endl;
        }
return 0;
}
```

Example

```
#include <iostream>
using namespace std ;
int main ()
{
        int i , j ;
        for ( i=1 ; i<=3; i++){
                for ( j=1; j<=3;j++){
```

```
            if ( i==1 && j ==1){
            cout << 'A' ;
            //cout << endl;
            }
            else if ( i >= j){
            cout <<'B';
            cout << " ";
            }
        }
    cout<<endl;
    }

        return 0;
}
```

```
#include <iostream>
using namespace std ;
int main ()
{
    int i , j ;
    for ( i=1 ; i<=5; i++){
        for ( j=1; j<=5;j++){
            if ( i+j <= 6)
```

```
                    cout <<"*";
                    else cout << " ";

               }
               cout<<endl;
          }
return 0;
}
```

Example

```
#include <iostream>
using namespace std;
int main()
{
    int n, i , j;
cout << "Enter number of rows: ";
cin >> n;
    for(i = 1; i <= n; i++)
      {
          for(j = 1; j <= i; j++)
             {
                cout << "*";
            }
         cout<<"\n";
      }

    for(i = n; i >= 1; i--)
      {
         for(j = 1; j <= i; j++)
            {
                cout << "*" ;
            }
         cout<<"\n";
      }
}
```

Example

```
#include <iostream>
using namespace std;
int main()
{
        int c=0;
        for(int i=1;i<=4;i++){
        for(int j=1;j<=7;j++) {
```

```
        if (i+j>=5 && j-i<=3){
                cout << (char) (c+'A');
                }
                else {
                cout << " ";


                }


        }
        cout << endl;
c++;
}


}
```