```
// Exercise 2.27 Solution: ex02_27.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   char symbol; // char read from user

   cout << "Enter a character: "; // prompt user for data
   cin >> symbol; // read the character from the keyboard

   cout << symbol << "'s integer equivalent is "
      << static_cast< int >( symbol ) << endl;
} // end main
```

==========

```
// Exercise 2.28 Solution: ex02_28.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int number = 0; // integer read from user

   cout << "Enter a five-digit integer: "; // prompt
   cin >> number; // read integer from user

   cout << number / 10000 << "   ";
   number = number % 10000;
   cout << number / 1000 << "   ";
   number = number % 1000;
   cout << number / 100 << "   ";
   number = number % 100;
   cout << number / 10 << "   ";
   number = number % 10;
   cout << number << endl;
} // end main
```

```
//////////////////////////////////////////////
// Fig. 1.14: fig01_14.cpp
// Using if statements, relational
// operators, and equality operators.
#include <iostream>

using std::cout;  // program uses cout
using std::cin;   // program uses cin
using std::endl;  // program uses endl

// function main begins program execution
int main()
{
   int num1;  // first number to be read from user
   int num2;  // second number to be read from user

   cout << "Enter two integers, and I will tell you\n"
        << "the relationships they satisfy: ";
   cin >> num1 >> num2;   // read two integers

   if ( num1 == num2 )
      cout << num1 << " is equal to " << num2 << endl;
```

```
   if ( num1 != num2 )
      cout << num1 << " is not equal to " << num2 << endl;

   if ( num1 < num2 )
      cout << num1 << " is less than " << num2 << endl;

   if ( num1 > num2 )
      cout << num1 << " is greater than " << num2 << endl;

   if ( num1 <= num2 )
      cout << num1 << " is less than or equal to "
           << num2 << endl;

   if ( num1 >= num2 )
      cout << num1 << " is greater than or equal to "
           << num2 << endl;
   return 0;   // indicate that program ended successfully
} // end function main
```

Check Whether Number is Even or Odd using ternary operators

```
#include <iostream>
using namespace std;
```

```
int main() {
  int n;

  cout << "Enter an integer: ";
  cin >> n;

  (n % 2 == 0) ? cout << n << " is even." :  cout << n << " is odd.";

  return 0;
}
```

Check Prime Number

```
/*
 * C++ Program to Check Prime Numbers
 */

using namespace std;
int main ()
{
    int num, i, count = 0;
    cout << "Enter the number to be checked : ";
    cin >> num;
```

```
    if (num == 0)
    {
        cout << "\n" << num << " is not prime";
        exit(1);
    }
    else   {
            for(i=2; i < num; i++)
                if (num % i == 0)
                    count++;
    }
    if (count > 1)
        cout << "\n" << num << " is not prime.";
    else
        cout << "\n" << num << " is prime.";
    return 0;
}
```

==========

Example 1: C++ if Statement

```
// Program to print positive number entered by the user
// If the user enters a negative number, it is skipped

#include <iostream>
```

```
using namespace std;

int main() {

  int number;

  cout << "Enter an integer: ";
  cin >> number;

  // checks if the number is positive
  if (number > 0) {
    cout << "You entered a positive integer: " << number << endl;
  }

  cout << "This statement is always executed.";

  return 0;
}
```

Example 2: C++ if...else Statement

```
// Program to check whether an integer is positive or negative
// This program considers 0 as a positive number
```

```
#include <iostream>
using namespace std;

int main() {

  int number;

  cout << "Enter an integer: ";
  cin >> number;

  if (number >= 0) {
    cout << "You entered a positive integer: " << number << endl;
  }
  else {
    cout << "You entered a negative integer: " << number << endl;
  }

  cout << "This line is always printed.";

  return 0;
}
```

==========

Example 3: C++ if...else...else if

```
// Program to check whether an integer is positive, negative or zero


#include <iostream>
using namespace std;


int main() {


  int number;


  cout << "Enter an integer: ";
  cin >> number;


  if (number > 0) {
    cout << "You entered a positive integer: " << number << endl;
  }
  else if (number < 0) {
    cout << "You entered a negative integer: " << number << endl;
  }
```

```
    else {
      cout << "You entered 0." << endl;
    }

    cout << "This line is always printed.";

    return 0;
}
```

If the number is greater than `0`, the code inside the `if` block is executed. If the number is less than `0`, the code inside the `else if` block is executed. Otherwise, the code inside the `else` block is executed.

Example : C++ Ternary Operator

```
#include <iostream>
#include <string>
using namespace std;

int main() {
  double marks;
```

```
  // take input from users
  cout << "Enter your marks: ";
  cin >> marks;

  // ternary operator checks if
  // marks is greater than 40
  string result = (marks >= 40) ? "passed" : "failed";

  cout << "You " << result << " the exam.";

  return 0;
}
```

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

    return 0;
}
```

```
// Ex. 2.25: ex_02_25.cpp
// What does this program print?
#include <iostream>

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

// function main begins program execution
int main()
{
   int row = 10;   // initialize row
   int column;     // declare column

   while ( row >= 1 ) {   // loop until row < 1
      column = 1;  // set column to 1 as iteration begins

      while ( column <= 10 ) {            // loop 10 times
         cout << ( row % 2 ? "<" : ">" );  // output
         ++column;                         // increment column

      } // end inner while

      --row;          // decrement row
```

```
      cout << endl;   // begin new output line

   } // end outer while

   return 0;  // indicate successful termination

} // end function main
```

```
// Ex. 2.24: ex02_24.cpp
// What does this program print?
#include <iostream>

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

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

   while ( count <= 10 ) {   // loop 10 times

      // output line of text
      cout << ( count % 2 ? "****" : "++++++++" )
```

```
         << endl;
      ++count;              // increment count
   }

   return 0;   // indicate successful termination

} // end function main
```

```
// Ex. 2.15: ex02_15.cpp
// What does this program print?
#include <iostream>

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

// function main begins program execution
int main()
{
   int y;          // declare y
   int x = 1;      // initialize x
   int total = 0;  // initialize total

   while ( x <= 10 ) {    // loop 10 times
```

```
      y = x * x;          // perform calculation
      cout << y << endl;  // output result
      total += y;         // add y to total
      ++x;                // increment counter x

   } // end while

   cout << "Total is " << total << endl;  // display result

   return 0;  // indicate successful termination

} // end function main
```

```
// Ex. 2.42: ex02_42.cpp
// What does this program print?
#include <iostream>

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

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

```
   int x,  // declare x
       y;  // declare y

   // prompt user for input
   cout << "Enter two integers in the range 1-20: ";
   cin >> x >> y;  // read values for x and y

   for ( int i = 1; i <= y; i++ ) {   // count from 1 to y

      for ( int j = 1; j <= x; j++ )  // count from 1 to x
         cout << '@';                 // output @

      cout << endl;                   // begin new line

   } // end outer for

   return 0;  // indicate successful termination

} // end function main
```

```
// 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
```