Chapter 8
Selection

Selection (Decision)

The second control logic structure is *selection*:

Selection – Choosing between two or more alternative actions.

Selection statements alter the sequential flow of instructions in a program. This is done using selection statements. When we make the computer choose between two alternatives we will define one of the alternatives as being true and the other as being false. This choice will be made using a *boolean expression.*

Boolean Expression – an expression that evaluates to either true or false

When boolean expressions are formed they must set up a relationship between items in the expression. This relationship is set up using a set of operators called the *relational operators*.

Relational Operators:

==	(equal)
<	(less than)
>	(greater than)
<=	(less than or equal)
>=	(greater than or equal)
!=	(not equal)

NOTE: The assignment operator is (=) but the relational operator is(= =). In other words, if you are assigning a value to a variable, use (=) but if you are asking whether two things are equal, use (= =). DO NOT COMPARE TWO FLOATING POINT VALUES FOR EQUALITY.

Examples of boolean expressions:

Expression	Result
`5 == 5`	`true`
`‘a’ < ‘c’`	`true`
`4 + 3 > 10`	`false`
`10 != 20`	`true`
`6 <= 6`	`true`
`5 >= 9`	`false`
`‘A’ < ‘Z’`	`true`
`‘a’ < ‘Z’`	`false (WHY?)`

Given two variables n1 and n2, assume n1 contains 5 and n2 contains 3. Evaluate each of the following boolean expressions:

n1 + 3 != n2 + 5

n1 * 2 <= n2 * 2

n1 < n2

n1 < n2 + 2

n2 <= n1

n2 + n1 + 3 == n1 + 6

The precedence order when the relational operators are added is as follows:

```
* /  %
+ -
< <=  > >=
==  !=
=
```

The bool constant `true` is stored internally as 1, and `false` is stored as 0. For this reason, given x = 5, and y = 7, the statement

```
    cout << (x <= y);
```

would output the value 1.

Exercises:

Explain whether the following statements are correct.

x != y   is equivalent to   y != x

a <= b   is equivalent to   b > a

x <= y   is equivalent to   y >= x

If the statement   a > b   is true, and the statement   b > c   is true, is the statement
    a > c   also true?

If the statement   a != b   is true, and the statement   a != c   is true, is the statement
    b != c   also true?

Given the following:

```
void main(void)
{
    int x, y, z;
    x = 5;
    y = 10;
    z = 2;

    cout << ( x * 2 < y) << endl;
    cout << ( x != y-2 ) << endl;
    cout << ( y > z ) << endl;
    cout << (2 * x + y > z * 10) << endl;
}
```

Show the output from the above code sample.

An expression using a relational operator (<, >, <=, >=, ==, !=) is called a
______________________ expression.

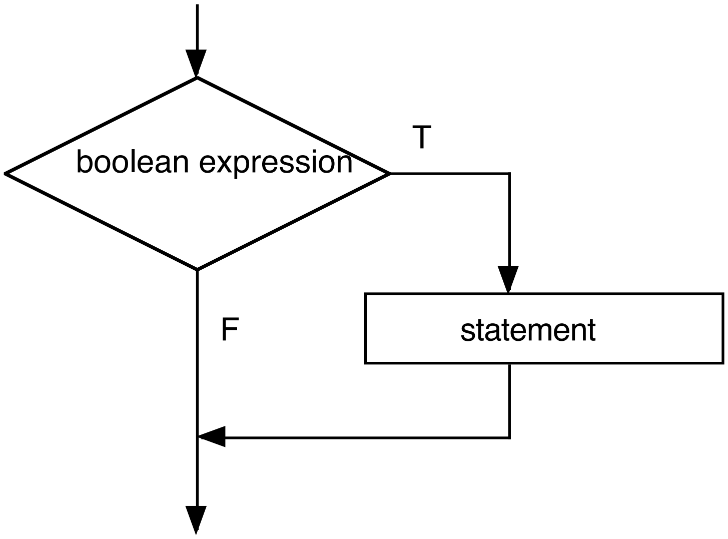
A boolean (relational) expression evaluates to either ______________ or ______________.

The numeric value of a boolean expression is 0 if the expression is __________________
and 1 if it is _________________________.

The C++ if and if-else Statements

The use of the *if* and *if-else* statements in a program allows the programmer to include all possible options in the program code and the computer to select the appropriate option during program execution.

Flowchart for the *if* statement

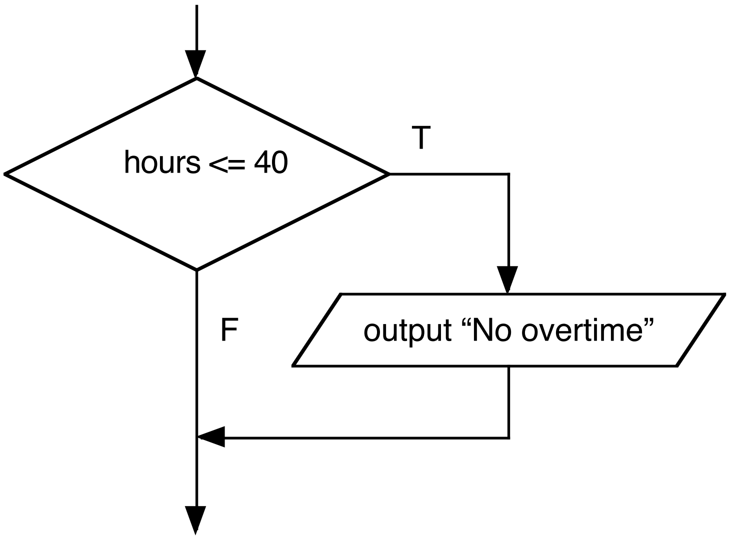


The boolean expression is evaluated and if it evaluates to true the statement inside the *if* is executed and the program continues to the next executable statement. If the expression evaluates to false, the statement inside the *if* is ignored and the program continues with the next executable statement.

GENERAL FORM of the if statement:

```
if (boolean expression)
   statement;
```

<u>**Any non-zero value is considered true by the *if* statement.**</u>



```
if (hours <= 40)
  cout << “No overtime” << endl;
cout << “Continue” << endl;
```

Given a value of 25 in the location hours, the program would output

```
No overtime
Continue
```

If location hours contained the value 50, the program would output

```
Continue
```

Consider the following:

```
if (hours <= 40)
  cout << “No overtime” << endl;
  cout << "You must work over 40 hours for overtime.\n";
cout << “Continue” << endl;
```

What would be output given the input values used above?

Recall, the general form for the if statement is

```
if(boolean expression)
    statement;
```

That means one statement only is executed if the expression is true. To make multiple statements appear to the compiler as though they are one statement, use a **compound statement**. A compound statement is one or more statements enclosed in French braces ({}).

```
if (hours <= 40)
{
  cout << “No overtime” << endl;
  cout << "You must work over 40 hours for overtime.\n";
}
cout << “Continue” << endl;
```

The program will now execute as desired. **It is a good practice to use French braces for all *if*, *if/else*, and looping statements.**

if Statement Exercises

1.) Draw the flowchart for a program that matches the following pseudocode:

```
- get a value for num1 from the user
- get a value for num2 from the user
- store the sum of num1 and num2 in location num1
- increment the location num2 by 5
- if num1 is equal to twice the value of num2 output a message stating that
  num1 is twice as large as num2
- output a message stating the program is over
```

2.) Comment on each code fragment below.

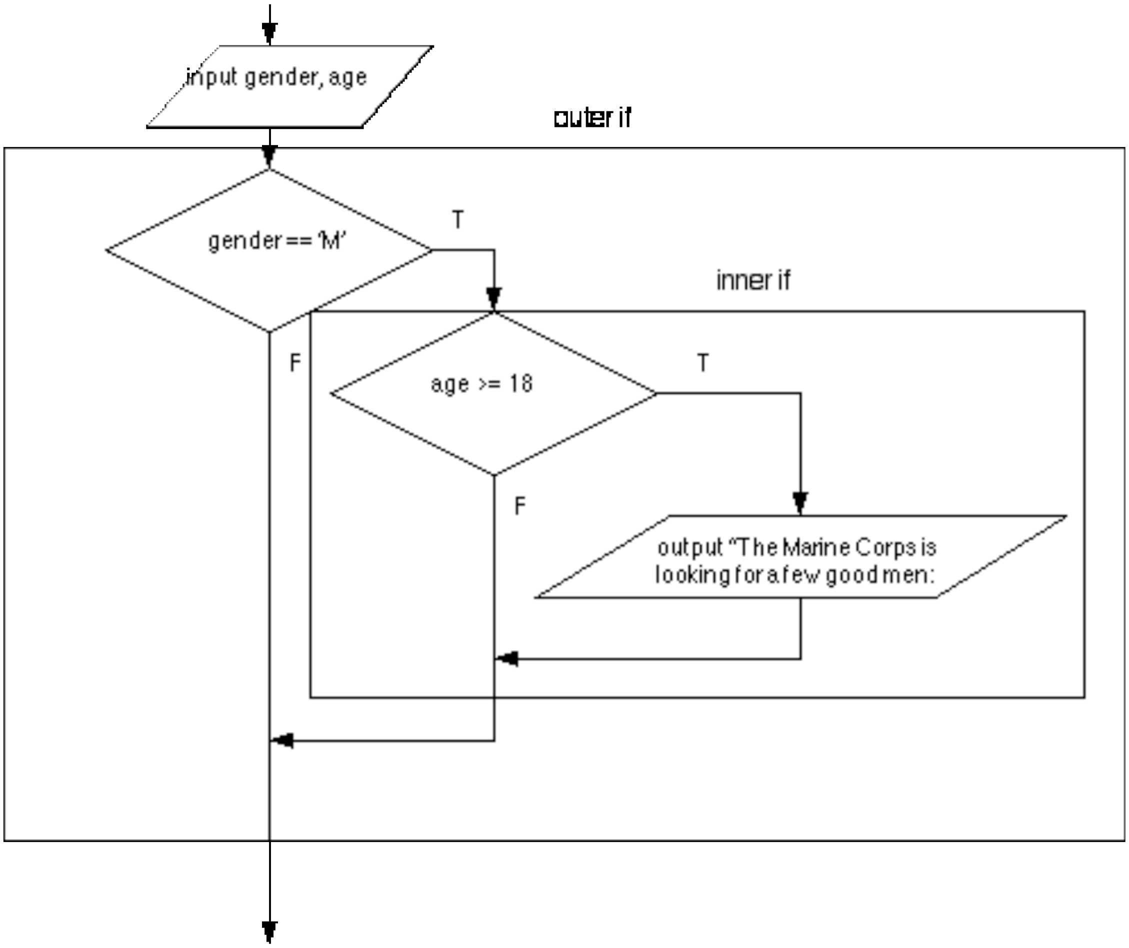
```
if(x == 5)
{
   cout << "value is five";
}
```

```
if( x <= y );
{
   cout << "y is greater than or equal to x";
}
```

```
if( x = -2 )
{
   cout << "x is negative 2";
}
```

Nested if Statements

A nested *if* is an *if* that contains another *if* within it. Suppose we wanted to identify males 18 and older. We could first determine whether the individual is male and if so, check to see whether the age requirement has been met.



```
 if(gender == 'M')
{
   if(age >= 18)
   {
      cout << "The Marine Corps is looking for a few good men."
   }
}
```

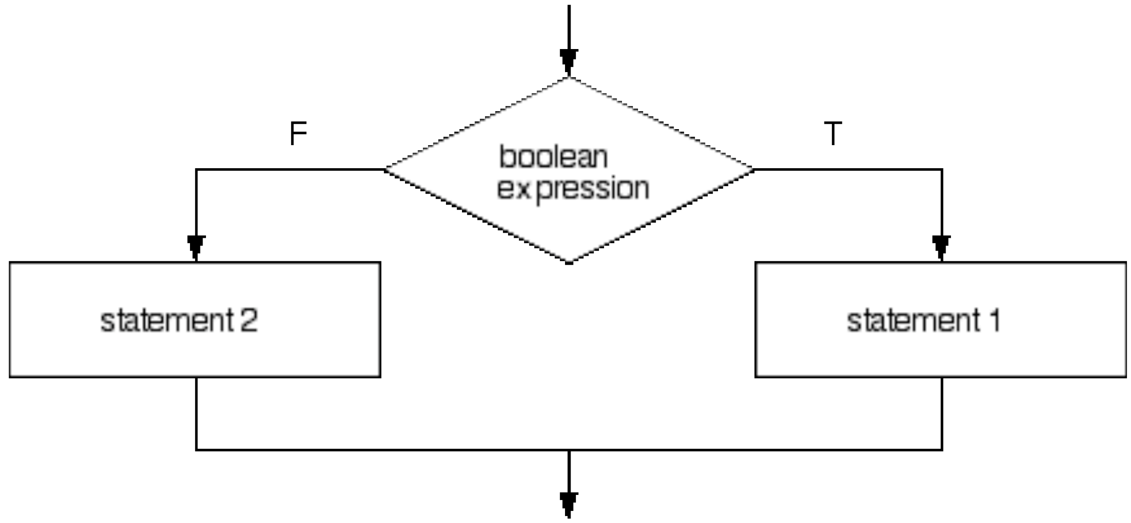
The if-else Statement

In the *if* statement, a condition was evaluated and if the condition evaluated to true, an instruction was executed and the program continued on, if the condition evaluated to false, the program simply continued on. The *if-else* statement assumes that some action will take place before the program continues. If the condition evaluates to true, do this, else do that.

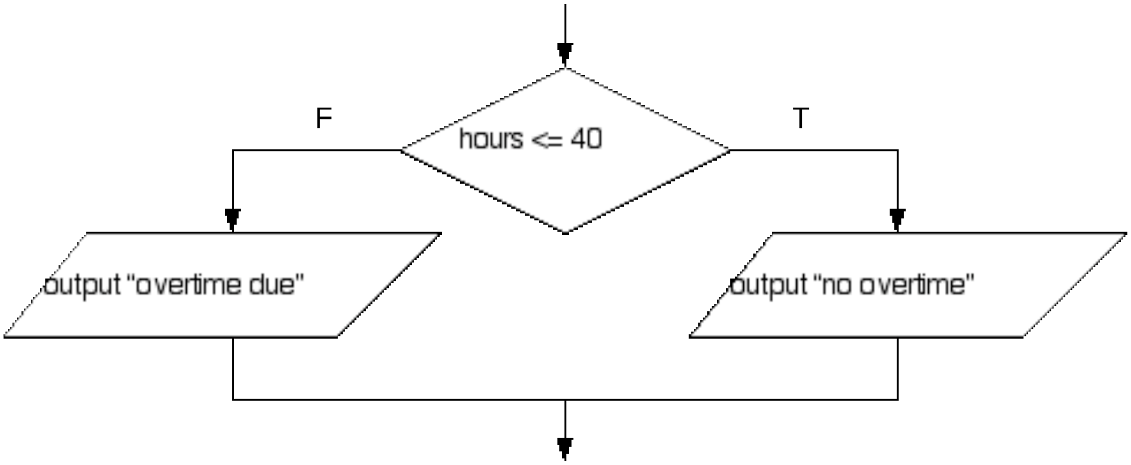
GENERAL FORM if-else Statement

```
if(boolean expression)
   statement 1;
else
   statement 2;
```

Flowchart for an if-else



Example:



C++ code:

```
if(hours <= 40)
{
  cout << “no overtime”;
}
else
{
  cout << “overtime due”;
}
```

Are the French braces necessary in the above example? If not, why are they used?

Examples:

```
if(num % 2 != 0 )
{
  cout << "Odd number";
}
else
{
  cout << "Even number";
}
```

```
if(speed > 65)
{
   cout << "Slow down";
}
else
{
   cout << "Speed is legal on I5";
}
```

Program Errors

There are three types of errors that can occur in programming. Syntax errors, run-time errors, and logic errors.

Syntax errors are grammatical errors and are found by the compiler. List several of these below.

Run-time errors are errors that occur after the program has successfully compiled and is executing. For example, your program might try to access a printer that is not turned on or attempted division by zero. Many run-time errors involve constructs beyond the scope of CS 1A.

Logic errors are the most difficult to find. While the program compiles without error, the results are erroneous. IF YOU SPENT AMPLE TIME IN THE DESIGN PHASE AND TESTED THOROUGHLY BEFORE YOU STARTED CODING, YOU SHOULD HAVE FEW, IF ANY LOGIC ERRORS.

It is important to note that attempting to fix errors during the implementation phase as opposed to the design phase greatly increases the chances of introducing even more errors into your program.

Comment on each of the following: If the code will generate an error, indicate the type.

```
if(speed > 65);
    cout << "Slow down";
else
    cout << "Speed is legal on I5";
cout << "\nDone";
```

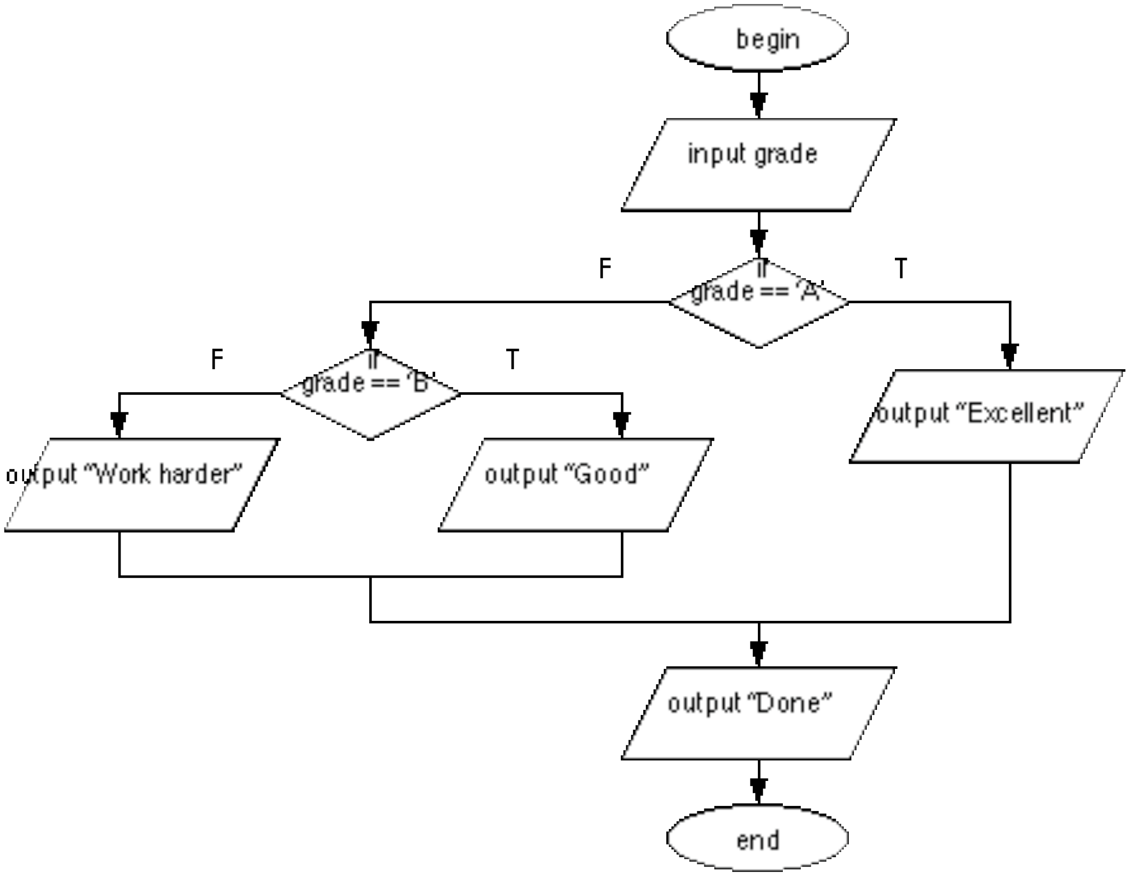
```
if(speed > 65)
    cout << "Slow down";
else;
    cout << "Speed is legal on I5";
cout << "\nDone";
```

```
if(speed > 65)
    cout << "Slow down";
    cout << "\nRelax, you'll live a longer and happier life\n";
else
    cout << "Speed is legal on I5";
```

```
if(speed > 65)
{
    cout << "Slow down";
    cout << "\nRelax, you'll live a longer and happier life\n";
}
else
    cout << "Speed is legal on I5";
```

```
if(speed > 65)
{
    cout << "Slow down";
    cout << "\nRelax, you'll live a longer and happier life\n";
}
else
    cout << "Speed is legal on I5";
    cout << "\nA lack of tickets is good for your insurance rates.\n";
cout << "I hope you learned something";
```

Nested if-else



```
cout << "Enter your grade (A - F): ";
cin >> grade;
if(grade == 'A')
{
    cout << "Excellent\n";
}
else
{
    if(grade == 'B')
    {
        cout << "Good\n";
    }
    else
    {
        cout << "Work harder\n";
    }
}
cout << "Done";
```

if-else Exercises

1. Flowchart and write a C++ program that will receive two integers from the user and output the larger of the two. We will assume the two values are not equal.

2. Flowchart and write a C++ program that will receive two integers from the user and store the numbers in locations called numerator and denominator. Calculate and output the result of the division if the denominator is not equal to zero otherwise output a message stating division by zero is an illegal operation.

3. Modify the program from question 1 to find the largest of three input integers. You will need a "nested" structure for this.

4. Flowchart and write a C++ program that will receive an employee name, hours worked for the week, and an hourly rate of pay from the user. Your program is to calculate an amount for the regular pay (hours up to and including 40), the overtime pay (time and one half for hours over 40), and the gross pay (the sum of the regular pay and overtime). Look for calculations that are always performed in the same manner and those that differ according to the hours worked. <u>You should not have any redundant code in your program</u>. Output the three calculated values at the end of your program.

5. Flowchart and write a C++ program that receives the grade point average for a student. If a student has a gpa of 3.5 or greater, output a message that they have made the Dean's List, otherwise see if their gpa is 2.0 or more, if so, output a message telling them their gpa is ok. Tell all other students to see their academic advisor.

6. Flowchart and write a C++ program that receives the current temperature as input. If the temperature is 80 degrees or more, output a message telling the user to go swimming, otherwise, if the temperature is 50 degrees or more, output a message to go running, otherwise stay inside.

Name ______________________________________
Due Date ___________________

Sales Commission Lab

Flowchart and write a program that accepts the first name of a sales person and their total sales for the month. Sales commissions are calculated as follows:

Monthly Sales	Commission Rate
1.00 - 1000.00	2%
1001.00 - 5000.00	5%
over 5000.00	10%

Sample Run:

```
Sales Associate:    HarryHardsell
Monthly Sales:      6500.00

Commission Due:     650.00
```

Turn in (IN THIS ORDER):

This sheet
Flowchart (template or software)
Listing of the code (SEE CHAPTER 5 FOR STYLE REQUIREMENTS)
3 sample runs (one for each commission rate)

Names ____

Due Date __________________

Sales Commission Part 2 - Homework

Modify the original lab assignment to incorporate the following:

1) Every sales person is paid a base salary of 1500.00 per month. Make the base salary a named constant.

2) The commission amount is cumulative. Example:
   An employee with 6500.00 in sales would receive a commission of 370.00
   (20.00 + 200.00 + 150.00)
   Their total pay would be 1870.00

Use the commission schedule from Part 1 of this assignment.

Monthly Sales	Commission Rate
1.00 - 1000.00	2%
1001.00 - 5000.00	5%
over 5000.00	10%

Turn in (IN THIS ORDER):

This sheet
Flowchart (template or software)
Listing of the code (SEE CHAPTER 5 FOR STYLE REQUIREMENTS)
3 sample runs (one for each commission rate

Mission Viejo Electric Company

Obtain the following integer values from the user:

- the date broken down into month, day and year (3 variables)
- the current meter reading
- the previous meter reading

Calculate an electric bill based on the following rate & consumption table:

Kilowatt Consumption	Cost
0 - 400 KWH	a flat rate of $8.50
401 – 1200 KWH	$8.50 (base rate) + 7.525 cents for each KWH over 400 (May – Sept.) or 6.575 cents for each KWH over 400 (all othe months)
over 1200 KWH	calculate the first 1200 KWH according to the formula given above plus 95% of the appropriate seasonal rate for the KWH over 1200

<u>Show the exact formulas (in C++ format) for all calculations</u>.

This program makes use a C++ boolean operator, the && (AND) operator. This operator allows you to ask multiple questions in one statement. For example if we were looking for an integer in the range of 1 – 10, the following expression would check for this condition.

```
num >= 1  &&  num <= 10
```

If both conditions evaluate to true, the entire expression is true. If one or both of the conditions evaluate to false, the entire expression is false.

Roots of a Quadratic Polynomial

Given the floating point variables a, b, c, root1, root2 and discriminant, flowchart and write a C++ program that determines whether the roots of a quadratic polynomial are real or imaginary. If the roots are real, find them and assign them to variables root1 and root2. If no real roots exist, print the message "no real roots". The formula for the solution to the quadratic equation is

$$\frac{-b \overset{+}{-} \sqrt{b^2-4ac}}{2a}$$

$\overset{+}{-}$ means "plus or minus" and indicates that there are two solutions to the equation; one in which the result of the square root is added to –b and one in which the result is subtracted from –b. The roots are real if the discriminant is not negative. If the discriminant is 0 then there is only one real root. The formula for the discriminant is

$$b^2 – 4ac$$

Review

1. Selection is one of the three control/logic structures. Define selection.

2. Describe the difference between the *if* and *if-else* statements in C++.

3. An expression that evaluates to true or false is called a _______________ expression.

4. The bool constant `true` is stored internally as ____ and `false` as ____.

5. One or more statements enclosed in French braces is called a __________ _____________________________.

6. The three types of errors that can occur in programming are __________, ______________________, and __________________.

7. List the C++ relational operators.