Chapter 5 Glossary. Conditions, logical expressions, & selection control structures

flow of control
The order in which statements are executed in a program.

control structure
Special statement that transfers control to a statement other than the one that physically comes next.

selection control structure
A branching control structure that allows you to choose between alternative actions.

Boolean expression
One that has the value true or false. An assertion is a Boolean expression.

relational operators
Operators that test a relationship between two values. The operations are (listed by their C++ symbols): ==, !=, >, <, >=, <=

logical operators
In C++, && (and), || (or), and ! (not). (see definitions below for each of these.)

The “and” (&&) logical operator
A && B is true if and only if A is true and B is true. That is, both expressions must be true.

The “or” (||) logical operator
A || B is true if either one (or both) of the expressions is true.

The “not” (!) logical operator
!A is true if A is false. !A is false if A is true.

short-circuit evaluation of Boolean expressions
The evaluation of a logical expression in left to right order with evaluation stopping as soon as the final truth value can be determined.

if-then-else (aka if-else) statement
A selection control structure that allows you to take one of two alternative actions depending on the truth of the “if” condition.

Format:

```cpp
if (someLogicalExpression)
    stmt a  \leftarrow \text{the if clause}
else
    stmt b  \leftarrow \text{the else clause}
```

if-then (aka if) statement
A selection control structure that allows you to take an action depending on the truth of the “if” condition. Like an in-else with a null else clause.

nested if
An if statement whose first instruction in its if clause is another if statement, as in—

```cpp
if (x > y)
    if (y == 3)
        ...
```

dangling else
An else clause for which it is not obvious which if it should be paired with. In the absence of braces, an else is always paired with the closest preceding if that doesn’t already have an if paired with it.

switch stmt  (see pp 462-466 in text)
A selection control structure used to select among many alternatives. Can only be used for particular types of data. (Refer to your class slides & notes). It is an alternative to an if-else chain.

testing the state of a stream
Using a C++ stream object in a logical expression as if if were a Boolean variable. The result is true if the last I/O operation on that stream succeeded; otherwise, it is false. Ex: inFile.open("someFile");

```cpp
if ( !inFile )
    cout << "Error"
    ...
```

return statement in “main”
Returns control to the operating system. It denotes the “exit status” of the program: 0 if everything is okay; another integer if not. If a non-zero value is returned, that indicates abnormal termination.

testing control statements
Using enough data sets when testing a program so that every branch is executed at least once. This is called minimum complete coverage.

white box testing
Allows us to see the program code while designing the tests, so that data values at the boundaries, and possibly middle values, can be tested.

black box testing
Testing as many allowable values as you can without having access to the program code. That is, the program is like a black box that you cannot see inside.

test plan
A document that specifies the test cases to try, the reason for each, and the expected output. You implement the test plan by verifying that the program outputs the predicted results.