2.2.4 Tests and Loops

C++ provides a conventional set of statements for expressing selection and looping. For example, here is a simple function that prompts the user and returns a Boolean indicating the response:

```
bool accept()
{
    cout << "Do you want to proceed (y or n)?\n";    // write question

    char answer = 0;
    cin >> answer;                                    // read answer

    if (answer == 'y') return true;
    return false;
}
```

To match the **<<** output operator ("put to"), the **>>** operator ("get from") is used for input; **cin** is the standard input stream. The type of the right-hand operand of **>>** determines what input is accepted, and its right-hand operand is the target of the input operation. The **\n** character at the end of the output string represents a newline (§2.2.1).

The example could be improved by taking ~~an **n**~~ (for "no") answer into account:

```
bool accept2()
{
    cout << "Do you want to proceed (y or n)?\n";    // write question

    char answer = 0;
    cin >> answer;                                    // read answer
```

```
        switch (answer) {
        case 'y':
                return true;
        case 'n':
                return false;
        default:
                cout << "I'll take that for a no.\n";
                return false;
        }
}
```

A **switch**-statement tests a value against a set of constants. The case constants must be distinct, and if the value tested does not match any of them, the **default** is chosen. If no **default** is provided, no action is taken if the value doesn't match any case constant.

Few programs are written without loops. For example, we might like to give the user a few tries to produce acceptable input:

```
bool accept3()
{
        int tries = 1;                                    true
        while (tries<4) {
                cout << "Do you want to proceed (y or n)?\n";      // write question
                char answer = 0;
                cin >> answer;                                     // read answer

                switch (answer) {
                case 'y':
                        return true;
                case 'n':
                        return false;
                default:
                        cout << "Sorry, I don't understand that.\n";
                        ++tries;    // increment
                }
        }
        cout << "I'll take that for a no.\n";
        return false;
}
```

Value ~~get~~ bigger if reach 4 it will stop the function

The **while**-statement executes until its condition becomes **false**.