Control Structures: Selection Statement

Chapter 3
B. Ramamurthy

Introduction

“Two roads divulged in a yellow wood,
And sorry I could not travel both....”

—— Robert Frost
From “The Road Not Taken”
Structured Programming

- Sequence
- Selection
- Repetition

Decision Statements (selection)

- How to compare data values?
  - Relational operators
- How to alter the sequence of program execution based on the result?
  - if.. else statements
- How to deal with multiple choices?
  - Switch statement
Example

We will use “assigning letter grade based on percentage points” as an example to illustrate selection statement.

<table>
<thead>
<tr>
<th>Percent Range</th>
<th>&gt;=90</th>
<th>&gt;=80 &lt; 90</th>
<th>&gt;=70 &lt; 80</th>
<th>&gt;=60 &lt; 70</th>
<th>&lt;50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter Grade</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>F</td>
</tr>
</tbody>
</table>

Relational Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>Less than ?</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than ?</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to?</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to?</td>
</tr>
<tr>
<td>==</td>
<td>Equal to?</td>
</tr>
<tr>
<td>!=</td>
<td>Not Equal to ?</td>
</tr>
</tbody>
</table>
Logical Operators

! not
&& and
|| or

These operators are used to combine more than one condition forming a complex condition.

if Statement

- An if statement allows a program to choose whether or not to execute a following statement.
- Syntax:
  if (condition)
  statement;
- Semantics:
  condition is a Boolean expression: Something that evaluates to True or False.
  If condition is true then execute the statement is executed.
The if statement : syntax

if(expression)
  statement;  //single statement executed
  //if expression is true

if(expression)
{
  //statements inside {} are
  //executed if expression is true
  statement1;
  statement2;
  ...
  statement n;
}

If -else Statement

An if-else statement allows a program to do one thing if a condition is true and a different thing if the condition is false.

Syntax:
if ( condition )
  statement1
else
  statement2

Statements to be executed for if and else can be a single statement or multiple statements enclosed in { }.
The if - else statement: syntax

```
if(expression)
    statement;
else
    statement;
if(expression)
    {
        statement block
    }
else
    {
        statement block
    }
```

The switch statement

```
switch(expression)
{
    case constant:
        statement(s);
        break;
    case constant:
        statement(s);
        break;
    /* default is optional*/
    default:
        statement(s);
}
```
The switch statement

- Expression must be of type integer or character
- The keyword case must be followed by a constant
- break statement is required unless you want all subsequent statements to be executed.

Practice!

Convert these nested if/else statements to a switch statement:

```cpp
if (rank==1 || rank==2)
    cout << "Lower division \n";
else
    { if (rank==3 || rank==4)
        cout << "Upper division \n";
        else
        { if (rank==5)
            cout << "Graduate student \n";
            else
            cout << "Invalid rank \n";
        }
    }
```
Practice Solution!

```c
switch(rank)
{
    case 1: case 2:
        cout << "Lower division \n";
        break;
    case 3: case 4:
        cout << "Upper division \n";
        break;
    case 5:
        cout << "Graduate student \n";
        break;
    default:
        cout << "Invalid rank \n";
}
//end switch
```

Summary

- In many applications, choices are to be made depending on some conditions related to the problem. Selection or decision structures are used to model such situations.
- C++ supports the implementation of “selection” through the “if” and “switch” statements. In this discussion we looked at various forms of selection statements.