Programming Fundamentals

Aamina Batool

If..else-if..else

```
if (score >= 90)
    cout << "The grade is A." << endl;
else if (score >= 80)
    cout << "The grade is B." << endl;
else if (score >= 70)
    cout << "The grade is C." << endl;
else if (score >= 60)
    cout << "The grade is D." << endl;
else
    cout << "The grade is F." << endl;
```

The switch Multiple-Selection Structure

- **`switch`**
  - Useful when variable or expression is tested for multiple values
  - Consists of a series of **`case`** labels and an optional **`default`** case
  - **`break`** is (almost always) necessary

```
switch (expression) {
        case val1:
                statement
                break;
        case val2:
                statement
                break;
          ....
        case valn:
                statement
                break;
        default:
                statement
                break;
}
```

```
if  (expression == val1)
        statement
else if (expression==val2)
        statement
        ....

else if (expression== valn)
        statement
else
        statement
```

```
int main()
{
	int x;
	cout<<"Enter the 1, 2, or 3 \n";
	cin>> x;

	swtich (x)
	{
		case 1:
			cout<<"case 1 statements\n";
			break;
		case 2:
			cout<<"case 2 statements\n";
			break;
		case 3:
			cout<<"case 3 statements\n";
			break;
		default:
			cout<<"invalid input\n";
			break;
	}
}
```

Switch Example

```
switch (grade)
{
case 'A':
     cout << "The grade point is 4.0.";
     break;
case 'B':
     cout << "The grade point is 3.0.";
     break;
case 'C':
     cout << "The grade point is 2.0.";
     break;
case 'D':
     cout << "The grade point is 1.0.";
     break;
case 'F':
     cout << "The grade point is 0.0.";
     break;
default:
     cout << "The grade is invalid.";
}
```

```
switch (score / 10)
{
case 0:
case 1:
case 2:
case 3:
case 4:
case 5:
     grade = 'F';
     break;

 case 6:
      grade = 'D';
      break;
 case 7:
      grade = 'C';
      break;
 case 8:
      grade = 'B';
      break;
 case 9:
 case 10:
      grade = 'A';
      break;
 default:
      cout << "Invalid test score." << endl;

 }
```

Nested Control Structures

- Suppose we want to create the following pattern

```
*
**
***
****
*****
```

- In the first line, we want to print one star, in the second line two stars and so on

Nested Control Structures (continued)

- Since five lines are to be printed, we start with the following for statement

```
for (i = 1; i <= 5 ; i++)
```

- The value of `i` in the first iteration is `1`, in the second iteration it is `2`, and so on
- Can use the value of `i` as limit condition in another for loop nested within this loop to control the number of starts in a line

Nested Control Structures (continued)

- The syntax is:

```
for (i = 1; i <= 5 ; i++)
{
    for (j = 1; j <= i; j++)
        cout << "*";
    cout << endl;
}
```

Nested Control Structures (continued)

- What pattern does the code produce if we replace the first for statement with the following?

```
for (i = 5; i >= 1; i--)
```

- Answer:

```
*****
****
***
**
*
```

Nested Control Structures

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

int main() {

    int rows = 5;
    int columns = 3;

    for (int i = 1; i <= rows; ++i) {
        for (int j = 1; j <= columns; ++j) {
            cout << "*  ";
        }
        cout << endl;
    }

    return 0;
}
```

Break inside nested loops

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

int main() {
    int weeks = 3, days_in_week = 7;

    for (int i = 1; i <= weeks; ++i) {
        cout << "Week:  " << i << endl;

        for (int j = 1; j <= days_in_week; ++j) {
            // break during the 2nd week
            if (i == 2) {
                break;
            }
            cout << "    Day:" << j << endl;
        }
    }
}
```

Continue statement - example

```
sum = 0;
cin >> num;

while (cin)
{
    if (num < 0)
    {
        cout << "Negative number found in the data." << endl;
        cin >> num;
        continue;
    }

    sum = sum + num;
    cin >> num;
}
```

Continue inside nested loops

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

int main() {
    int weeks = 3, days_in_week = 7;

    for (int i = 1; i <= weeks; ++i) {
        cout << "Week:  " << i << endl;

        for (int j = 1; j <= days_in_week; ++j) {
            // continue if the day is an odd number
            if (j % 2 != 0) {
                continue;
            }
            cout << "    Day:" << j << endl;
        }
    }
}
```