

Programming 1: Tutorial 4

Introduction

For this tutorial you are going to the concept of controlling the flow of the program, something key in making games interesting.

You can either over write previous question solutions with new ones or you can have a project for each of the exercises – don't forget to make them blank applications.

Hopefully you remember the basic layout of a C++ program?

If not here is a refresher, basic C++ programs like the ones you have been asked to do here have the following layout:

```
#include <iostream>
// include statements up here

int main()
{
    // block of code between the { and } brackets

    return 0;
}
```

It may also help if you go through a design stage for each exercise where you plan and puzzle through the logic before actually coding.

To complete these exercises you may need to reference the notes, either online or use the printed ones provided. Doing this isn't a bad thing; most programmers have to look up the syntax to make sure there code is correct.

There is sample code at the end of the tutorials, typing and running this code may help you when it comes to solving the exercises.

It is also important to understand what is happening with the questions, rather than just typing it into the compiler. If you look at it, run it and don't understand then just ask.

Question 1: Logical Equals (==) Versus Assignment Equals (=)

Do the following statements produce the same results and, which one is correct?

```
if (age = 10)    // Is this the correct Equals to use?
{
    cout << "You are 10 ! " << endl;
}
```

```
if (age == 10)  // Or is this the correct Equals to use?
```

```
{
    cout << "You are 10 ! " << endl;
}
```

What would happen? Why?

Question 2: Round Brackets

Do the following compound conditions produce the same results?

```
if ((age == 10) || (age == 15)) && (weight == 5))
{
    cout << "Condition satisfied" << endl;
}
```

```
if ((age == 10) || ((age == 15) && (weight == 5)))
{
    cout << "Condition satisfied" << endl;
}
```

if Statements

For the following exercises you will need to use if or if-statements to solve the problems.

Exercise 1: More age related programming

This builds on the previous 'user age' programs we have programmed. Write a program that asks for the user's age, checks to see if they are old or young (your choice of what that value) is then prints out there age and a suitable message.

Exercise 2: Report Card Generator

Ask the user to enter grades for various "subjects" (the number is up to you). Check to see if the grades are valid (between 0 and 100), then print out if the user got an A, B, C, D, or F in his subjects. Divide the letters however you like. Use an if-else-if statement to find out which grade they receive. The grade boundaries are also up to you.

Example:

Biology: <user enters value>

Maths: <user enters value>

You Achieved:

Biology A

Maths U

switch Statements

For the following exercises you will need to use switch to solve the problems.

Exercise 3: Multi-Function Calculator

Ask the user for 2 numbers, and then ask them to select from a numbered list of mathematically operations.

As Print a suitable message if they enter an invalid value.

Depending on the operator selected (+, -, *, and /) do the operation on the numbers and print out the result including the equation with the proper operator. Remember to check for situations such as division by zero. You'll have to use floating-point numbers here as well.

Example:

```
Please enter two numbers
```

```
Number 1: <user enters value>
```

```
Number 2: <user enters value>
```

```
Choose an arithmetic operation:
```

```
+, -, /, *
```

```
Result = <Number1> < arithmetic operation> <Number 2>
```

Sample Programs

Assignment

```
// A short program with examples of assignment statements
#include <iostream.h>

int main()
{
    //Data declarations
    int i, j, k, m, n, surprise;

    float pi, c, z;

    //Assignment Statements
    i = 1;           // i is assigned the value 1

    j = 2;           // j is assigned the value 2

    k = 13/4;        // The result of 13/4 is 3 which is assigned to k

    i = i + j;       // The result of i + j is 3 which is assigned
                    // to i (the old value of i is overwritten.
    m = i * k;       // The result of i * k is 9 which is assigned
                    // to m
    n = m % j;       // The remainder of m divided by j is 1 which
                    // is assigned to n
    pi = 3.142;     // The value 3.142 is assigned to the float pi

    z = pi + 1;     // The value of pi + 1 is 4.142 which is
                    // assigned to z
    surprise = z;   // What happens when a floating point number is
                    // assigned to an integer identifier?

    return 0;
}
```

If you wish to check the output of the variables, add `cout` statements.

If-else

```
#include <iostream>

using namespace std;

int main()
{
    char c;

    cout << "Enter a character : ";
    cin >> c;

    if (c == ' ')
    {
        cout << "Blank" << endl;
    }
    else if (c >= '0' && c <= '9')
    {
        cout << "Digit" << endl;
    }
    else if (c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z')
    {
        cout << "Letter" << endl;
    }
    else
    {
        cout << "Something else !" << endl;
    }

    return 0;
}
```

Switch

```
#include <iostream>

using namespace std;

int main()
{
    char input;

    cout << "a. Play game \n\n";
    cout << "c. Load game \n\n";
    cout << "d. Play multiplayer \n\n";
    cout << "e. Exit \n\n";

    cin >> input;

    switch (input)
    {
        case 'a': case 'A':
        {
            cout << "playing \n\n";
            break;
        }
        case 'c': case 'C':
        {
            cout << "loading \n\n";
            break;
        }
        case 'd': case 'D':
        {
            cout << "multiplayer \n\n";
            break;
        }
        case 'e': case 'E':
        {
            cout << "You have choosen to quit \n\n";
            break;
        }
        default:
        {
            cout<<"Error, bad input, quitting \n\n";
        }
    }

    cout << "raar";
}

return 0;
}
```

Another switch example:

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string code = "password";
    char letter = 0;

    int sLength = (int)code.length();

    for(int i = 0; i < sLength; i++)
    {
        letter = code.at(i);

        switch (letter)
        {
            case 'a':
                cout << 'z';
                break;
            case 's':
                cout << 'x';
                break;
            case 'p':
                cout << 'j';
                break;
            case 'w':
                cout << 'l';
                break;
            case 'o':
                cout << 'q';
                break;
            case 'r':
                cout << 'f';
                break;
            case 'd':
                cout << 'g';
                break;
            default:
                cout << letter;
                break;
        }
    }

    return 0;
}
```