UEE1302(1066) F12: Introduction to Computers and Programming

Flow of Control (I) - Selection

What you will learn from Lab 3

In this laboratory, you will understand how to use selection (`if` and `switch`) to perform flow control.

TASK 3-0 : WORKING WITH UNIX/LINUX MACHINES

- ✓ Click PieTTY to start
  - ➢ Specify the machine name/ip as one of the following:
    - ✧ **140.113.170.180**
  - ➢ Select **SSH**, not `Telnet(BBS)`,
- ✓ Authentication
  - ➢ Username: Student ID #
  - ➢ Password: UEE1302-Student ID#
- ✓ Compile the program from your file
  - ➢ `g++ <filename>.cpp -o <program>`
    - ✧ (Example) `>g++ hello.cpp -o hello` ↵
  - ➢ Check if any error occurs. If there is one, the message will be printed on screen. Try to correct the error(s) according to the system responses.
- ✓ Execute the program
  - ➢ If the compilation succeeds, run the program you designated under the current directory.
  - ➢ `>./<program>`
    - ✧ (Example) `>./hello` ↵

TASK 3-1 : FLOW CONTROL STATEMENT - `if`

- ✓ Compile `lab3-1.cpp` and execute program `lab3-1`.
- ✓ Try to conclude the difference between "==" and "!=".

```
//File: lab3-1.cpp
#include <iostream>
using namespace std;
↵
int main()
{
    int x, y;
    x = 3;
```

```
    y = 5;
    if (x == 3)
    {
        cout << "x = " << x << endl;
    }
    if (y != 5)
    {
        cout << "y = " << y << endl;
    }
    return 0;
}
```

- ✓ Modify program `lab3-1.cpp` to `lab3-2.cpp` and execute program `lab3-2`.
  - ➢ Use command "cp" to copy file. (Ex. >`cp lab3_1.cpp lab3_2.cpp`)
- ✓ Try to conclude the difference between "==" and "=".

```
//File: lab3-2.cpp
#include <iostream>
using namespace std;

int main()
{
    int x, y;
    x = 3;
    y == 5;
    if (x = 5)
    {
        cout << "x = " << x << endl;
        cout << "y = " << y << endl;
    }
    return 0;
}
```

- ✓ Compile `lab3-3.cpp` and execute program `lab3-3`. Record your result.

```
//File: lab3-3.cpp
#include <iostream>
using namespace std;

int main()
```

```
{
    int x;
    cout << "Please enter one number?" << endl;
    cin >> x;
    if ((x >= 5) && (x < 2))
    {
        cout << "HIT" << endl;
    }
    else
    {
        cout << "x = " << x << " is not in range" << endl;
    }

    return 0;
}
```

- Change "`if ((x >= 5) && (x < 2))`" to "`if ((x >= 5) || (x < 2))`" and explain the result

TASK 3-2 : FLOW CONTROL STATEMENT – `if` Chain

✓ Compile `lab3-4`.cpp and execute program `lab3-4`. Record your result.

```
//File: lab3-4.cpp
#include <iostream>
using namespace std;
int main()
{
    float score;
    char grade;
    cout << "Enter your score :" << endl;
    cin >> score;
    if ( score >= 80.0)
        grade = 'A';
    else if  (score >= 70.0)
        grade = 'B';
    else if (score >= 60.0)
        grade = 'C';
    else
        grade = 'F';
    cout << "You get level " << grade << endl;
    return 0;
}
```

TASK 3-3 : FLOW CONTROL STATEMENT – multiple `if`

- ✓ Modify program `lab3-4.cpp` to `lab3-5.cpp` and execute program `lab3-5`.
- ✓ Compare the different between "`if` chain" and "multiple `if`"

```
//File: lab3-5.cpp
#include <iostream>
using namespace std;
int main()
{
    float score;
    char grade;
    cout << "Enter your score :" << endl;
    cin >> score;
    if (score >= 80.0)
        grade = 'A';
    if (score >= 70.0)
        grade = 'B';
    if (score >= 60.)
        grade = 'C';
    if (score < 60.0)
        grade = 'F';
    cout << "You get level " << grade << endl;
    return 0;
}
```

  - ➢ Does the previous result belong to the functionality of switch? If not, please correct the wrong place(s) and verify it.

TASK 3-4 : FLOW CONTROL STATEMENT - `switch`

- ✓ Compile `lab3-6.cpp` and execute program `lab3-6`. Record your result.

```
//File: lab3-6.cpp
#include <iostream>
using namespace std;

int main()
{
    char sym;
    cout << "Enter a symbol:" << endl;
    cin >> sym;
    switch (sym)
    {
        case 'a':
            cout << "The symbol is one 'a'." << endl;
        case 'b':
```

```
            cout << "The symbol is one 'b'." << endl;
        case 'c':
            cout << "The symbol is one 'c'." << endl;
        default:
            cout << "The symbol is not 'a', 'b', or 'c'." << endl;
    }
↵
    return 0;
}
```

- Does the previous result belong to the functionality of switch? If not, please correct the wrong place(s) and verify it.
- Assume that the default case is removed. If the input does not belong to the set `{a,b,c}`, what can you conclude?
  - ✧ What is the role of "default" case here?

TASK 3-5 : EXERCISES

✓ Design a system where two players can play the finger-guessing game ('s' for scissors, 'r' for rock, 'p' for paper). The required format is shown as follows.

```
>./ex3-1 ↵
Enter s, r, p for player 1: r ↵
Enter s, r, p for player 2: p ↵
Player 2 win!
>
```

```
>./ex3-1 ↵
Enter s, r, p  for player 1: s ↵
Enter s, r, p  for player 2: s ↵
Two players tie!
>
```

✓ Design a book-buying system. In the system, the customer could choose which book he/she wants to buy and the quantity of books. The required format is shown as follows.

```
>./ex3-2 ↵
Welcome to the book-buying system!!
The following are the list and the cost:
(A) Computer Science: An Overview      $30
(B) Absolute C++       $20
(C) C++ How to Program   $40
Please enter A, B, C to choose the book:
A ↵
Now enter the quantity of books:
3 ↵
The total cost of item (A) is $90. Thanks for your coming.
>
```

```
>./ex3-2 ↵
Welcome to the book-buying system!!
The following are the list and the cost:
(A) Computer Science: An Overview      $30
(B) Absolute C++       $20
(C) C++ How to Program   $40
Please enter A, B, C to choose the book:
D ↵
Sorry, item (D) doesn't on our list.
>
```

- Note that the lower-case letter is allowed. (Ex. Choice could be 'a' or 'A')
- Use '\t' between item and cost.;