

## C LAB WORKSHEET 8

### C & C++ Selection: if, if-else, if-else-if Part 1

Items in this page:

1. The C & C++ conditional statement, a selection.
2. The if, if-else construct and flowcharts
3. Activities, questions and answers.
4. Tutorial reference that should be used together with this worksheet are: [C & C++ program control 1](#) and [C & C++ program control 2](#).

- In this lab worksheet we will deal with another C/C++ program control, a selection. The if, if-else construct and its variation used in C/C++ when we need to make a **selection** from the given options based on the certain **condition**. For example:

```
if (x == 3)
    printf("The if condition is fulfilled!\n");
printf("The if condition is not fulfilled!\n");
```

- From the code snippet, if  $x = 3$ , then the first printf() will be executed else the second printf() will be executed. So a simplest if form is shown below.

```
if (condition)
{
    statement(s);
}
next_statement;
```

- If the **condition** is true, the **statement(s)** is executed else the **next\_statement** is executed. Similar to the loop discussed in the previous worksheet, if there is more than one line of code, you need to use the curly braces. Let me make the previous example as a full working program.
- Create a project named **ifprog**. Then add C++ source file named **ifprogrsrc** to the project. Don't forget to set your project to be compiled as C code. Then try the following C program.

```
#include <stdio.h>

int main(void)
{
    int x;
    printf("Enter an integer: \n");
    // you can try using scanf() for older compilers...
    scanf_s("%d", &x, 1);
    if(x == 3)
    {
        printf("x = %d\n", x);
        printf("The if condition is fulfilled!\n");
    }
    printf("This is the next statement, after the if body...\n");
    return 0;
}
```

- If 3 is entered the output is shown below. The  $x == 3$  is **true** then the if statement is executed.

- If other than 3 is entered, the output is shown below. The if statement is skipped because  $x == 3$  is **false**. Take note about the different between = and ==.

- By expanding the simplest if statement, the if-else constructs make C/C++ more flexible in selecting options based on several conditions. For example:

```
#include <stdio.h>
```

```
int main(void)
{
    int x;
    printf("Enter an integer: \n");
    scanf_s("%d", &x, 1);
    if(x == 3)
    {
        printf("x = %d\n", x);
        printf("The if condition is fulfilled!\n");
    }
    else
    {
        printf("x = %d\n", x);
        printf("The if condition is not fulfilled!\n");
    }
    return 0;
}
```

- If 3 is entered, the output is shown on the right.

- If other than 3 is entered, the output is shown on the right.

- The general form is given below.

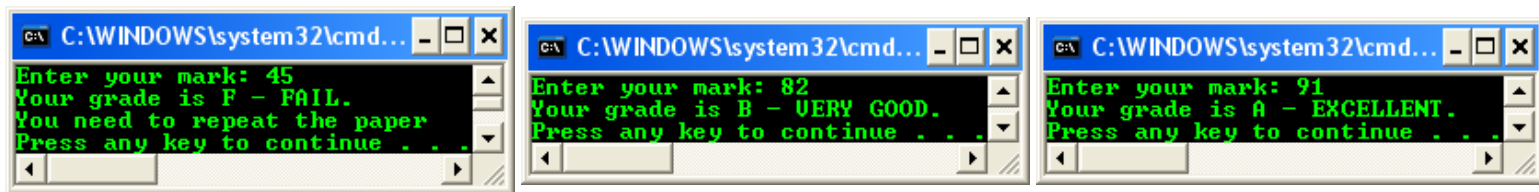
```
if (condition)
{
    statement 1;
}
else
{
    statement 2;
}
next_statement;
```

- Again, by expanding the **if-else** statement, we can build a program with any number of conditions and the respective options.

```
#include <stdio.h>

int main(void)
{
    int x;
    printf("Enter your mark: ");
    scanf_s("%d", &x, 1);
    if(x <= 50)
    {
        printf("Your grade is F - FAIL.\n");
        printf("You need to repeat the paper\n");
    }
    else if(x <= 60)
        printf("Your grade is E - CONDITIONAL PASS.\n");
    else if(x <= 70)
        printf("Your grade is D - PASS.\n");
    else if(x <= 80)
        printf("Your grade is C - GOOD.\n");
    else if(x <= 90)
        printf("Your grade is B - VERY GOOD.\n");
    else if(x <= 100)
        printf("Your grade is A - EXCELLENT.\n");
    return 0;
}
```

- Program input and output samples are given below.



- The general form is shown below.

```
if (condition1)
{
    statement 1;
}
else if (condition2)
{
    statement 2;
}
else if(...)
{
    statement ...;
}
next_statement;
```

- Another variation is checking **several conditions** for an option. In this if construct, every **if** must be matched with the **else**. For example:

```
#include <stdio.h>

int main(void)
{
    int year, train;
    char curr_post;
    printf("Are you qualified for a Manager promotion?\n");
    printf("Enter the number of year: ");
    scanf_s("%d", &year, 1);
```

```

printf("Enter the number of training received: ");
scanf_s("%d", &train, 1);
printf("Enter your current post(J - Junior, S - Senior, A - Asst. Manager): ");
// don't forget the space before %c else the execution will stop prematurely...
scanf_s(" %c", &curr_post);
if(year >= 10)
    if(train >= 5)
        if(curr_post == 'A' || 'a')
            printf("Qualified to be promoted to Manager!\n");
        else
        {
            printf("Become Assistant Manager first.\n");
            printf("Need more training.\n");
            printf("Need more experience.\n");
        }
    else
    {
        printf("Need more training.\n");
        printf("Need more experience.\n");
    }
else
    printf("Need more experience.\n");
return 0;
}

```

- A sample input and output is shown below.

- In this example, to be promoted to a manager post, all the three `if` statements must be true.
- In [Win32 Tutorial](#) you will notice that the `if` statement is very useful for simple C/C++ code troubleshooting. We can test/verify/check by skipping certain line of codes to verify whether the error is generated by the skipped code.

| [Main](#) |< [C & C++ while & do-while loop 2](#) | [C & C++ if and if-else 2](#) >| [Site Index](#) | [Download](#) |

The C Selection `if`, `if-else`, `if-else-if`, `break`, conditional/ternary operator and `switch-case-break`: [Part 1](#) | [Part 2](#) | [Part 3](#) | [Part 4](#) | [Part 5](#) | [Part 6](#)