Condition Statement

CSC10012 - Fundamentals of Programming

Lecturer: Dr. Le Ngoc Thanh, lnthanh@fit.hcmus.edu.vn
Teaching Assistant: Mr. Le Nhut Nam, lnnam@fit.hcmus.edu.vn

Prepare your submission

① **For multiple-choice and essay questions**: Present your work to the PDF file whose name is ⟨StudentID⟩ .pdf, where StudentID is your student number. The following guide you how to treat with different types of questions

- Single-choice questions: Each choice is indicated by the circle ○. Pick the most correct choice and explain for every other choice (i.e., why you did not choose it)
- Multiple-choice questions: Each choice is indicated by the square □ Pick one or many correct choices and explain for every other choice
- Essay questions: Give a comprehensive answer that covers all aspects of the problem

② **For C/C++ programming questions**: Present your source codes in C/C++. Do not use any additional libraries/open sources without permission

- Prepare a separate folder for each question. Name the folder as ⟨StudentID_⟩ ⟨i⟩ where StudentID is your student number and i is the question index. For example, 24120001_1, 24120001_2, etc.
- Put all corresponding files (i.e., files of extensions *.cpp, *.h, and data files, etc.) to each folder. Do not include the intermediate files generated by the Visual Studio compiler. Do not include the execution files since they are easily affected by viruses.
- Programming questions sometimes require brief explanation. In those cases, present your arguments to the report file described above.

③ Solutions for multiple-choice and essay questions will be provided after the submission deadlines, while for programming questions, solutions can be alternatively algorithms or C/C++ source codes.

④ Assignment will have too many questions. Don't worry about this. Please complete as many questions as you can.

⑤ Students contact TA through Moodle forum or email (Mr. Le Nhut Nam: lnnam@fit...).

1 Multiple-choice questions and essay questions (10pts)

1.1. How many choices are possible when using a single if-else statement?

○ 1

● 2 a single if-else statement have to choices: once when condition is true, false

○ 4

○ None of these

1.2. How many choices are possible when using a single switch statement?

○ 1

○ 2

○ 4

● It depends on specific cases.

1.3. What does each of the following code fragment write to the console?

Code fragment A

```
int sum = 14;
if (sum < 20)
    cout << "Under";
else
    cout << "Over";
cout << "the␣limit";
```

○ Under

● Under the limit

○ Over

○ Over the limit

Code fragment B

```
int sum = 14;
if (sum < 20)
    cout << "Under";
else {
    cout << "Over␣";
    cout << "the␣limit";
}
```

● Under

○ Under the limit

○ Over

○ Over the limit

Code fragment C

```
int sum = 94;
if (sum < 20){
    cout << "Under";
    cout << "the␣limit";
}
else {
    cout << "Over";
    cout << "the␣limit";
}
```

○ Under

○ Under the limit

○ Over

● Over the limit

1.4. What does each of the following code fragment write to the console?

Code fragment A

```
int sum = 7;
if (sum > 20)
    cout << "You␣win␣";
else
    cout << "You␣lose␣";
cout << "the␣prize";
```

○ You win the prize

○ You win

● You lose the prize

○ You lose

Code fragment B

```
int sum = 21;
if (sum == 20)
    cout << "You␣win␣";
else
    cout << "You␣lose␣";
cout << "the␣prize";
```

○ You win the prize

○ You win

● You lose the prize

○ You lose

Code fragment C

```
int sum = 21;
if (sum != 20)
    cout << "You␣win␣";
else
    cout << "You␣lose␣";
cout << "the␣prize";
```

● You win the prize

○ You win

○ You lose the prize

○ You lose

1.5. What value is assigned to discount in each of the following code?

Code fragment A

```
double discount;
char code = 'C';
switch (code){
    case 'A':
        discount = 0.0;
        break;
    case 'B':
        discount = 0.1;
        break;
    case 'C':
        discount = 0.2;
        break;
    default:
        discount = 0.3;
}
```

○ 0.0

○ 0.1

● 0.2

○ 0.3

Code fragment B

```
double discount;
char code = 'C';
switch (code){
    case 'A':
        discount = 0.0;
        break;
    case 'B':
        discount = 0.1;
        break;
    case 'C':
        discount = 0.2;
        break;
    default:
        discount = 0.3;
}
```

○ 0.0

○ 0.1

● 0.2

○ 0.3

Code fragment C

```
double discount;
char code = 'X';
switch (code){
    case 'A':
        discount = 0.0;
        break;
    case 'B':
        discount = 0.1;
        break;
    case 'C':
        discount = 0.2;
        break;
    default:
        discount = 0.3;
}
```

○ 0.0

○ 0.1

○ 0.2

● 0.3

1.6. Is it always possible to convert an if-then statement to a switch-case statement? And vice versa? Explain your answers.

It's not always possible to convert an if-else statement to a switch-case statement, and vice versa. Because if-else statement's condition are Logical Operators while switch-case statement are just equality operator.

1.7. What value is placed in choice?

```
int a = 5, b = 10, c =1 5 ;
choice = a > b && a > c ? a : (b > c ? b : c) ;
```

0 && a > c

0 -----------> do b > c ? b : c

0 ----> c

=> choice = 15

1.8. Consider the following statement

```
int y = !(3 > x && 3 <= 5 || 12 < 5) ? 7 : 9;
```

If x = 2, how many comparisons will be made to determine the value of y? Briefly explain each comparison.

! (1 && 1 || 12 < 5)

! (1 || 12 < 5)

! (1)

0

=> y = 9

need 2 comparisons

1.9. The following code fragment may include some syntax/semantic/ logic errors. Point out such errors and suggest how to fix them.

```
if numNeighbors >= 3 || numNeighbors = 4
    ++numNeighbors;
    cout << "You␣are␣dead!";
else
    --numNeighbors;
```

syntax error: the logic expression, numNeighbors = 4
semantic error: 2 statement of if
logic error: numNeighbors = 4

```
if (numNeighbors >= 3)
{
   ++numNeighbors;
   cout << "You are dead!";
}
else
   --numNeighbors;
```

1.9. Consider the following code fragment. Assume that the variables `doesSignificantWork`, `makesBreakthrough`, and `nobelPrizeCandidate` are all of bool type.

```
if (doesSignificantWork){
    if (makesBreakthrough)
        nobelPrizeCandidate = true;
    else
        nobelPrizeCandidate = false;
}
else if (!doesSignificantWork)
    nobelPrizeCandidate = false;
```

It is possible to improve the above code fragment by using fewer if statement(s). How? Moreover, the code fragment can also be rewritten using only a single statement. How?

```
if (doesSignificantWork)
      nobelPrizeCandidate = makesBreakthrough;
else
      nobelPrizeCandidate = doesSignificantWork;
```

```
nobelPrizeCandidate = doesSignificantWork && makesBreakthrough;
```