0. Casual Discussion, Warm-up Questions, and Lecture Demo Exercises

Casual Discussion, Warm-up Questions

Spend 5 minutes for the questions in this part. See how much you can finish. 😊

A. (i) If you find difficulties in **take-home exercises**, you should:

<table>
<thead>
<tr>
<th>Put ☑ or ☐ for each choice.</th>
</tr>
</thead>
<tbody>
<tr>
<td>read the question again</td>
</tr>
<tr>
<td>check hints from Guided-exercises or redo the Guided-ex again</td>
</tr>
<tr>
<td>read Helena's email and check course web</td>
</tr>
<tr>
<td>revision and reading (lecture notes and textbook)</td>
</tr>
<tr>
<td>try changing the code here and there and wait for luck</td>
</tr>
<tr>
<td>next time I should start doing exercises earlier (not last minute)</td>
</tr>
</tbody>
</table>

(ii) True/False [Put ☑ or ☐ below.]

- "My code is better than you because I use less variables."
- "My code is better than you because my logic is simple and clear. It is easy to read and maintain."
- "My code is better than you because it is short and tricky."

B. True/False [Put ☑ or ☐ below.]

- Every "cin <<" pauses for my input until I press <enter>.

Ref. Lab02 - page 1 - CASE 1.

C. The code below makes use of **setw** and **setfill**. Your task: Complete ① and ②:

```cpp
#include <iostream>
#include <________>  // ①
using namespace std;

void main()
{
    cout << "12345678901234567890" << endl;
    cout << "====================" << endl;
    cout << setw(4) << 89 << setw(10) << "Hello" << 3 << endl;
    cout << setfill('0') << setw(4) << 67;
    cout << setw(5) << 45 << setfill(' ') << setw(6) << 23 << endl;
}
```

By default:
- numbers are right-aligned and
- text are also right-aligned

What will be shown in the output?

```
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
= = = = = = = = = = = = = = = = = = = = = =
P r e s s a n y k e y t o c o n t i n u e .. .
```

Output:
D. Given the code below, trace by hands and fill in the table.

```cpp
.. void main()
{  
  int a, b, c, d;
  a = -10;
  a = a + 1;
  b = a;
  b++;
  c = -b;
  d = c % 3;
  cout << a << " " << b << " " << c << " " << d << endl;
}
```

**Values of variables:**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>after <code>a=-10</code></td>
<td><code>-10</code></td>
<td>[value undefined]</td>
<td>[value undefined]</td>
<td>[value undefined]</td>
</tr>
<tr>
<td>after <code>a=a+1</code> :</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>after <code>b=a</code> :</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>after <code>b++</code> :</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>after <code>c=-b</code> :</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>after <code>d=c%3</code> :</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>after <code>d+1</code> :</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The output will be:  

F. (i.) Fix the error of the following using **full braces** and improve it with **correct indentations**.

```cpp
if (choice=='w')
  cout << "Welcome!";  
  cout << endl;
else
  if (choice=='h')
    cout << "Hello!";  
    cout << endl;
  else
    cout << "Invalid choice!";  
    cout << endl;
```

(ii.) "If the executed sequence involves only 1 statement, then `{}` are optional in most cases."

Your task: **Rewrite** the code with **no braces**.

For your reference - ASCII table

<table>
<thead>
<tr>
<th>Characters</th>
<th><code>0</code></th>
<th><code>1</code></th>
<th>...</th>
<th><code>9</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>48</td>
<td>49</td>
<td></td>
<td>57</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Characters</th>
<th><code>A</code></th>
<th><code>B</code></th>
<th>...</th>
<th><code>Z</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>65</td>
<td>66</td>
<td></td>
<td>90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Characters</th>
<th><code>a</code></th>
<th><code>b</code></th>
<th>...</th>
<th><code>z</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>97</td>
<td>98</td>
<td></td>
<td>125</td>
</tr>
</tbody>
</table>
Lecture Demo Exercises

Q1. Consider the code:

(i) What is the structure of the above program code as seen by the compiler?  
□ An if-statement nested inside an if-else statement  
□ An if-else statement nested inside an if-statement  
□ An if-statement followed by an if-else statement  
□ None of the above.

(ii) The code can’t behave correctly in some cases. 
Describe one such case: when temperature is __________.

(iii) How to correct it?

(iv) "In order to make our code easy to follow, we should attempt to handle the easiest case first."
How would you further improve the code above?

Q2. Consider the code:

# include <iostream>
using namespace std;

void main()
{
    int s;
    cout << "Input your score: ";
    cin >> s;

    if (s >= 85)
        cout << "A";
    else if (s >= 70) // line 12
        cout << "B"; // line 13
    else if (s >= 60) // line 14
        cout << "C"; // line 15

    if (s >= 60)
        cout << " (PASSED)" << endl;
    else
        cout << "F (Failed)" << endl;
}

(i) Complete the output in the cases below:

<table>
<thead>
<tr>
<th>Input your score: 70</th>
<th>Input your score: 59</th>
</tr>
</thead>
<tbody>
<tr>
<td>Press any key to continue . . .</td>
<td>Press any key to continue . . .</td>
</tr>
</tbody>
</table>

Write down the expected output here.
(ii) If we exchange lines 12-13 and lines 14-15 like:

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

void main()
{
    int s;
    cout << "Input your score: ";
    cin >> s;
    if (s >= 85)
        cout << "A"; //line 14
    else if (s >= 60)  //line 15
        cout << "C";
    else if (s >= 70)  //line 12
        cout << "B";  //line 13
    if (s >= 60)
        cout << " (PASSED)" << endl;
    else
        cout << "F (Failed)" << endl;
}
```

Any problem? Show it below:

```
Input your score: 70

Press any key to continue . . .
```