



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
Faculty of Engineering

UNIVERSITI TEKNOLOGI MALAYSIA

TEST 2 (DEBUGGING)

SEMESTER I 2020/2021

SUBJECT CODE : SECJ/SCSJ1013
SUBJECT NAME : PROGRAMMING TECHNIQUE I
YEAR/COURSE : 1 (SECJ/ SECV/ SECB/ SECR/ SECP)
TIME : 20:10 – 20:50 MYT (40 minutes)
DATE : 4th JANUARY 2021 (Monday)

INSTRUCTIONS TO THE STUDENTS:

- Please read the *General Guidelines for the Programming Technique I Test 2* that is shared in Telegram's Group and/ or provided in UTM e-learning
- Read the problem and instructions carefully.
- You are given **FORTY MINUTES** to complete the test inclusive of the submission of your program (**25 minutes to answer** the question and **15 minutes to submit** the answer).
- Your program must follow the input and output as required in the text and shown in the examples. You must test the programs with (but not limited to) all the input given in the examples.

IMPORTANT NOTES:

- All the **COMMENT STATEMENTS** in the submitted program **WILL NOT BE EVALUATED**.

SUBMISSION PROCEDURE:

- Only the source code is required for the submission and the source code's file shall be named as follows: *Name_matricesNo_section.cpp* (i.e. *AinaAli_A20EC018_01.cpp*).
- You do not need to compress the file.
- Submit the source code file via the **UTM's e-learning system**.

SAMPLE SET 1

Question

[35 Marks]

You are given a C++ program (**Test2Q1.cpp**) with 11 errors (syntax errors and/ or logical errors, if any). The program is developed to calculate the total price of an item after the inclusion of tax. It has three (3) user-defined functions as listed below:

Function Name	Description
get_Price	To get price from the user. This function will keep asking the user to enter valid input if the price value entered is not a valid one.
get_Type	To get item type from the user. This function will repeatedly ask the user to enter a valid integer number (1, 2, or 3) to represent item type.
tax	To calculate the tax based on the item's original price and type. Read the comment section before tax function definition for details about the percentage of tax to be imposed on the item based on its type.

The **main** function of the program has **cout** statements to display the item price that was previously entered by the user via the **get_Price** function and item price after a call to **tax** function was made. The **tax** function should be able to update the item's price that includes the tax imposed on the item based on its type. You are required to debug the errors, compile, and run the program. You are **NOT ALLOWED** to **remove** any statements in the program. You are only allowed to **update** the statements provided in the program and add a new statement(s) if absolutely necessary.

The program should produce the outputs as in **Figure 1**. **Note:** The values in **bold** are input by the user.

```
1 //Test2Q1.cpp
2 #include <iostream>
3 using namespace std;
4
5 // function prototypes
6 float get_Price();
7 int get_Type();
8 void tax(float, int);
9
10 // start main function
11 int main() {
12     float price = 0; // should > 0
13
14     int type = 0; // 1 -> Electronic
15                 // 2 -> Sports & Outdoor
16                 // 3 -> Vaccine
17
18     price = get_Price();
19     type = get_type();
20
21     cout << "Price before tax = " << price << "\n\n";
22     tax(price, type);
23     cout << "Total price after tax = " << price << '\n';
```

```

24
25     return 0;
26 }
27
28 // start new user-defined functions
29 void get_Price() {
30     float p = 0;
31
32     while (p < 0) {
33         cout << "Item price (RM): ";
34         cin >> p;
35     }
36
37     return p;
38 }
39
40 float get_Type() {
41     int t == 0;
42
43     do {
44         cout << "\n";
45         cout << "1 -> Electronic\n";
46         cout << "2 -> Sports & Outdoor\n";
47         cout << "3 -> Vaccine\n";
48         cout << "Item type (1,2,3): ";
49         cin >> t;
50     } while (t > 1 || t < 3);
51
52     cout << "\n";
53     return t;
54 }
55
56 // tax is based on item types as follows:
57 // 1 -> Electronic => 5%
58 // 2 -> Sports & Outdoor => 3%
59 // 3 -> Vaccine => 1%
60 void tax(float p, int t) {
61     float tax = 0;
62     switch (t) {
63         case 1: tax = p * 5 / 100; break;
64         case 2: tax = p * 3 / 100;
65         case 3: tax = p * 1 / 100;
66     }
67
68     cout << "Tax cost = " << tax << "\n";
69     p += tax;
70 }

```

Sample Output for Program Execution 1

```

Item price (RM): -1
Item price (RM): 100

1 -> Electronic
2 -> Sports & Outdoor
3 -> Vaccine
Item type (1,2,3): 4

1 -> Electronic

```

```
2 -> Sports & Outdoor
3 -> Vaccine
Item type (1,2,3): 2

Price before tax = 100

Tax cost = 3
Total price after tax = 103
```

Sample Output for Program Execution 2

```
Item price (RM): 0
Item price (RM): 200

1 -> Electronic
2 -> Sports & Outdoor
3 -> Vaccine
Item type (1,2,3): -5

1 -> Electronic
2 -> Sports & Outdoor
3 -> Vaccine
Item type (1,2,3): 1

Price before tax = 200

Tax cost = 10
Total price after tax = 210
```

Sample Output for Program Execution 3

```
Item price (RM): -5
Item price (RM): 300

1 -> Electronic
2 -> Sports & Outdoor
3 -> Vaccine
Item type (1,2,3): 0

1 -> Electronic
2 -> Sports & Outdoor
3 -> Vaccine
Item type (1,2,3): 3

Price before tax = 300

Tax cost = 3
Total price after tax = 303
```

Figure 1: The example of outputs

SAMPLE SET 2

Question

[35 Marks]

You are given a C++ program (**Test2Q1.cpp**) with 11 errors (syntax errors and/ or logical errors, if any). The program is developed to determine COVID-19 risk status. It has three (3) user-defined functions as listed below:

Function Name	Description
display_Question	To display the question text.
yes_No	To get an answer from the user. This function will repeatedly ask the answer from the users until it gets a valid input ('y' or 'n').
get_Status	It will return integer values either 0, 1, 2, or 3 that respectively represent the GREEN, YELLOW, ORANGE, or RED status. The parameters (red_zone , close_contact , fever) were previously set to 'y' or 'n' by a series of calls to the yes_No function.

The **main** function of the program has a series of calls to **display_Question** and **yes_No** functions inside a loop control structure. You are required to debug the errors, compile, and run the program. You are **NOT ALLOWED** to **remove** any statements in the program. You are only allowed to **update** the statements provided in the program and add a new statement(s) if absolutely necessary.

The program should produce the outputs as in **Figure 1**. **Note:** The values in **bold** are input by the user.

```
1 //Test2Q1.cpp
2 #include <iostream>
3 using namespace std;
4
5 // function prototypes
6 void display_Question();
7 void yes_No(char);
8 int get_Status(char, char);
9
10 // start main function
11 int main() {
12     char red_zone, close_contact, fever; //two possible character values
13                                         //only: 'y' -> yes, 'n' -> no
14     int status; // 0 -> GREEN, 1 -> YELLOW, 2 -> ORANGE, 3 -> RED
15
16     for (int i = 1; i < 3; i++) {
17         display_question(i);
18
19         if (i == 1)
20             yes_No(red_zone); // set red_zone either 'y' or 'n'
21         else if (i == 2)
22             yes_No(close_contact); // set close_contact either 'y' or 'n'
23         else
24             yes_No(fever); // set fever either 'y' or 'n'
25     }
```

```

26
27 // get risk status based on red_zone, close_contact, fever parameters
28 status = get_Status(red_zone, close_contact, fever);
29
30 cout << "Your Covid-19 risk status is ";
31 switch (status) {
32     case 0: cout << "GREEN"; break;
33     case 1: cout << "YELLOW"; break;
34     case 2: cout << "ORANGE";
35     case 3: cout << "RED";
36 }
37 cout << "\n";
38
39 return 0;
40 }
41
42 // start new user-defined functions
43 void display_Question(int q) {
44     switch (q)
45     case 1: cout << "Living in red zone?\n"; break;
46     case 2: cout << "Have a close contact with Covid-19 patient?\n";
47             break;
48     case 3: cout << "Body temperature >= 38 degrees Celcius?\n";
49 }
50 }
51
52 void yes_No(char ans) {
53     do {
54         cout << "Please enter your answer (y / n): ";
55         cin >> ans;
56     } while (ans == 'n' && ans == 'y');
57
58     cout << '\n';
59 }
60
61 void get_Status(char rz, char cc, char f) {
62     int s = 0;
63
64     if (rz == 'y') s++;
65     if (cc == 'y') s++;
66     if (f == 'y') s++;
67
68     return s;
69 }

```

Sample Output for Program Execution 1

```

Living in red zone?
Please enter your answer (y / n): n

Have a close contact with Covid-19 patient?
Please enter your answer (y / n): n

Body temperature >= 38 degrees Celcius?
Please enter your answer (y / n): t
Please enter your answer (y / n): a
Please enter your answer (y / n): n

```


<p>Your Covid-19 risk status is GREEN</p>
<p>Sample Output for Program Execution 2</p> <p>Living in red zone? Please enter your answer (y / n): n</p> <p>Have a close contact with Covid-19 patient? Please enter your answer (y / n): y</p> <p>Body temperature >= 38 degrees Celcius? Please enter your answer (y / n): p Please enter your answer (y / n): y</p> <p>Your Covid-19 risk status is ORANGE</p>
<p>Sample Output for Program Execution 3</p> <p>Living in red zone? Please enter your answer (y / n): q Please enter your answer (y / n): y</p> <p>Have a close contact with Covid-19 patient? Please enter your answer (y / n): n</p> <p>Body temperature >= 38 degrees Celcius? Please enter your answer (y / n): n</p> <p>Your Covid-19 risk status is YELLOW</p>
<p>Sample Output for Program Execution 4</p> <p>Living in red zone? Please enter your answer (y / n): q Please enter your answer (y / n): y</p> <p>Have a close contact with Covid-19 patient? Please enter your answer (y / n): s Please enter your answer (y / n): y</p> <p>Body temperature >= 38 degrees Celcius? Please enter your answer (y / n): t Please enter your answer (y / n): b Please enter your answer (y / n): y</p> <p>Your Covid-19 risk status is RED</p>

Figure 1: The example of outputs

SAMPLE SET 3

Question

[35 Marks]

You are given a C++ program (**Test2Q1.cpp**) with 11 errors (syntax errors and/ or logical errors, if any). The program is developed to calculate the total price of an item after the inclusion of postage cost. It has three (3) user-defined functions as listed below:

Function Name	Description
get_Weight	To get weight from the user. This function will keep asking the user to enter valid weight input (1 to 3 kg).
get_Price	To get price from the user. This function will keep asking the user to enter valid input if the price value entered is not a valid one.
postage	To calculate the postage cost based on the item's weight. Read the comment section before postage function definition for details about postage cost imposed on the item based on its weight.

The **main** function of the program has **cout** statements to display the item price that was previously entered by the user via the **get_Price** function and item price after a call to **postage** function was made. The **postage** function should be able to update the item's price that includes postage cost imposed on the item based on its weight. You are required to debug the errors, compile, and run the program. You are **NOT ALLOWED** to **remove** any statements in the program. You are only allowed to **update** the statements provided in the program and add a new statement(s) if absolutely necessary.

The program should produce the outputs as in **Figure 1**. **Note:** The values in **bold** are input by the user.

1	<code>//Test2Q1.cpp</code>
2	<code>#include <iostream></code>
3	<code>#define MAX_WEIGHT 3.0</code>
4	<code>using namespace std;</code>
5	
6	<code>// function prototypes</code>
7	<code>float get_Price;</code>
8	<code>float get_Weight();</code>
9	<code>void postage(float, float);</code>
10	
11	<code>// start main function</code>
12	<code>int main() {</code>
13	<code> float weight == 0; // should not more than 3 kg</code>
14	<code> float price = 0; // should > 0</code>
15	
16	<code> weight = get_weight();</code>
17	<code> price = get_Price();</code>
18	
19	<code> cout << "Price not including postage = " << price << "\n\n";</code>
20	
21	<code> // The postage function should be able to calculate and set price</code>
22	<code> // which including postage cost</code>
23	<code> postage(price, weight);</code>

```

24
25     cout << "Total price including postage = " << price << '\n';
26
27     return 0;
28 }
29
30 // start new user-defined functions
31 float get_price() {
32     float p = 0;
33
34     cout << "\n";
35     while (p == 0) {
36         cout << "Item price (RM): ";
37         cin >> p;
38     }
39
40     return p;
41 }
42
43 void get_Weight() {
44     float w = 0;
45
46     do {
47         cout << "Item weight (not more than 3 kg): ";
48         cin >> w;
49     } while (w <= 0 && w > MAX_WEIGHT);
50
51     return w;
52 }
53
54 // weight more than 2 kg postage cost is RM 10
55 // weight 1 - 2 kg postage cost is RM 7
56 // weight less than 1 kg postage cost is RM 4
57 void postage(float p, float w) {
58     float post = 0;
59
60     if (w >= 2) {
61         post = 10;
62     } else if (w > 1) {
63         post = 7;
64     } else {
65         post = 4;
66     }
67
68     cout << "Postage cost = " << post << "\n";
69     p += post;
70 }

```

Sample Output for Program Execution 1

Item weight (not more than 3 kg): **-1**
Item weight (not more than 3 kg): **3.1**
Item weight (not more than 3 kg): **2**

Item price (RM): **0**
Item price (RM): **100**
Price not including postage = 100

Postage cost = 7
Total price including postage = 107

Sample Output for Program Execution 2

Item weight (not more than 3 kg): **2.9**

Item price (RM): **-1**
Item price (RM): **200**
Price not including postage = 200

Postage cost = 10
Total price including postage = 210

Sample Output for Program Execution 3

Item weight (not more than 3 kg): **3.5**
Item weight (not more than 3 kg): **0.5**

Item price (RM): **0**
Item price (RM): **-5**
Item price (RM): **5**
Price not including postage = 5

Postage cost = 4
Total price including postage = 9

Figure 1: The example of outputs