TUTORIAL 4-3 (SECJ1013)

PROGRAMMING TECHNIQUE 1

SECTION 04 & 07, SEM 1, 2024/2025

Q#1

a) Given the definition of a function named **`exchange`** in Figure 1.

```
1  void exchange(int &x, int y)
2  {
3   int temp = x;
4   x = y;
5   y = temp;
6  }
```

Figure 1

Determine whether each of the following calls to the function **`exchange`** is valid. If it is, write the output printed by the code segment. Otherwise, give the reason why the call is invalid.

i.
```
int p = 1, q = 2;
exchange(p,q);
cout << "p:" <<p << ",q:" << q;
```

ii.
```
int r = 3;
exchange(r,10);
cout << "r:" << r;
```

iii.
```
int s = 4;
exchange(10,s);
cout <<"s:"<<s;
```

iv.
```
int t = 5, u = 6;
exchange(t,t+u);
cout << "t:" << t << ", u:" << u;
```

b) Using appropriate parameter passing (i.e., by reference or value or a combination of both), write the definitions for the following functions. You also need to determine the parameters that should be used by each function. Then write an example code that shows a correct call to the function.

i. **`biggest`**. This function finds the biggest value from two numbers.

ii. **`increment`**. This function increases the value of a variable by a specified amount. For example, if the value of a variable, `num`, is 10 and we increase it by 5, then its value becomes 15.

Note: *Provide your answer in a table as follow*

No	Function definition	Example of function call
i.		
ii.		

Q#2

Program 1 is given below. What is the output produced by this program?

```
1   //Program 1
2   #include <iostream>
3   using namespace std;
4
5   void sum(int a, int& b){
6      a = a + b;
7      b = a + b;
8      cout << "a =" << a << ", b =" << b << endl;
9   }
10
11  void sum(int a, int b, int c){
12     a++;
13     b++;
14     c = a + b;
15     cout<<"a =" <<a <<", b =" <<b <<", c =" <<c <<endl;
16  }
17
18  int sum(int a, double b) {
19     return a+b;
20  }
21
22  void copy (int& a, int& b){
23     a *= 2;
24     b *= 2;
25     cout << "a =" << a << ", b =" << b << endl;
26  }
27
28  int main(){
29     int x = 2, y =3 , z=0; sum(x, x);
30     z = sum(y,1);
31     cout << "x =" << x << " , z =" << z << endl;
32     copy(x,y);
33     sum(x,y,z);
34     cout << "x =" << x << ", y =" << y << ", z =" << z;
35     return 0;
36  }
```

Q#3

Program 2 is given below. What is the output produced by this program?

```
1   //Program 2
2   #include <iostream>
3   using namespace std;
4
5   int z = 0;
6
7   int f (int k, int m) {
8      static int n = 10;
9      n = n + k;
10     m = n;
11     return k + n;
12  }
13
14  int main() {
15     int k = 3, m = 5;
16     cout << k << " " << m << " " << f(k,m) << endl;
17     z = f(k,m);
18     cout << k << " " << m << " " << z << endl;
19     return 0;
20  }
```

Q#4

A length can be measured in miles, or yards and feet. A mile is equivalent to 5280 feet and a yard is equivalent to 3 feet.

a) Write a function named **`yardToMile`** to convert measurements in **`yards`** and **`feet`** to measurements in **`miles`**. The function accepts input of **`yards`** and **`feet`** (whole numbers) and returns an output of the total number of **`miles`** (a floating-point value). Use appropriate data types, parameter passing and return mechanisms in creating your function. Assume that this function is called from the main function with different sets of values and prints the output in **`miles`**.

b) Then, write a call statement to invoke the function defined in (a) to convert the length of 8 yards and 2 feet into miles.

Q#5

Based on the code segment below, write C++ code statements to perform each of the following instructions. Note that, each question is dependent.

```
1   char d1[10] = "disease";
2   char d2[10] = "autism ";
3   char d3[15];
4   int len;
```

a) Concatenates **`d1`** string to the end of **`d2`** string.

_____

b) Copy the resulting string in (a) into **`d3`**.

_____

c) Store the length of **d3** into **len**.

_____

d) Converts **d3** string to uppercase by filling in the blanks in the code segment below with an appropriate statement.

```
for (int x = 0; _________________; x++)
  ____________________________;
```

Q#6

Given three (3) overloaded function below:

```
1   void product(int &item, double &price) {
2     item = 100;
3     price = price * item;
4   }
5
6   void product(double price) {
7      price = price + 700.0;
8   }
9
10  void product(int &item) {
11     item = item + 5;
12  }
```

Determine the values of **i** and **p** after the execution of each following function calls. Note that, each question is independent. Assume the initial values of the variables for each question are **int i = 50** and **double p = 100.5**.

	Answers	
	i	**p**
(a) **product(i);**		
(b) **product(i, p);**		
(c) **product(p);**		

Q#7

Given two (2) function definitions:

(a) **increment**	(b) **increment**

void increment(float &r, int n) {
  float score = 10.0;
 	
  for(int i = 1; i <= n; i++)
  score = score + 2;
 	
  r = score;
}
	void increment(float &r, int n) {
	  float static score = 10.0;
	
	  for(int i = 1; i <= n; i++)
	  score = score + 2;
	
	  r = score;
	}

Based on code segment in **`main`** function, write an output for **Line 3** and **Line 6** in **Table 1** below. Note that, each question (a) **`increment`** and (b) **`increment`** is independent.

Table 1

Line	`main` function	Output (a) `increment`	Output (b) `increment`
1	`loop = 1;`		
2	`increment(result, loop);`		
3	**`cout << result << endl;`**		
4	`loop = 2;`		
5	`increment(result, loop);`		
6	**`cout << result << endl;`**		

Q#8

Given four functions named **`functionW`**, **`functionX`**, **`functionY`** and **`functionZ`**, in the **Program 3**. Write the output of the program.

```
1   //Program 3
2   #include <iostream>
3   using namespace std;
4
5   void functionW(int &w1, int w2 = 10) {
6       w1 = w2;
7       w2 = w1 * w2;
8   }
9
10  int functionX (int x1 = 8, int x2 = 3) {
11      x1 = x2;
12      x2 = x1 * x2;
13      return x1;
14  }
15
16  void functionY(int y) {
17      y = 10;
18  }
19
20  void functionZ(int z []) {
21      z[0] += 10;
22  }
23
24  int main()
25  {
26      int n1 = 2, n2 = 5;
27      int nums [] = {1};
28
29      functionW(n2, n1);
30      cout << "n1 = " << n1 << endl;
31      cout << "n2 = " << n2 << endl;
32
33      functionW(n1);
34      cout << "n1 = " << n1 << endl;
35      cout << "n2 = " << n2 << endl;
36
```

```
37      functionX(n2, n1);
38      cout << "n1 = " << n1 << endl;
39      cout << "n2 = " << n2 << endl;
40
41      n1 = functionX();
42      cout << "n1 = " << n1 << endl;
43      cout << "n2 = " << n2 << endl;
44
45      functionY(nums[0]);
46      cout << "nums[0] = " << nums[0] << endl;
47
48      functionZ(nums);
49      cout << "nums[0] = " << nums[0] << endl;
50
51      return 0;
52  }
```

Q#9

Complete Program 4 below. Trace the output of each math function and brief what it does.

```
1   //Program 4
2   #include <iostream>
3   //Write an appropriate directive here
4
5   using namespace std;
6   int main() {
7      cout << "\n 1. sqrt(9) is " << sqrt(9);
8      cout << "\n 2. pow(5,3) is " << pow(5,3);
9      cout << "\n 3. ceil(2.5) is " << ceil(2.5);
10     cout << "\n 4. floor(2.5) is " << floor(2.5);
11     cout << "\n 5. ceil(2.1) is " << ceil(2.1);
12     cout << "\n 6. floor(2.1) is " << floor(2.1);
13
14     return 0;
15  }
```

Q#10

What is the output of the following C++ statements? What is the difference between the results in line 5 and 6?

```
1   double x = 781.2856789;
2
3   cout << fixed << setprecision(5);
4   cout << x << endl;
5   cout << (floor(x * 100 + 0.5)/100) << endl;
6   cout << (int(x * 100 + 0.5)/100) << endl;
```

Q#11

Program 5 uses several predefined functions from several libraries. Identify the functions and their corresponding include files. Determine the output of the program.

```
1   //Program 5
2   #include <iostream>
3   #include <cctype>
4   #include <cstring>
5   #include <cstdlib>
6   using namespace std;
7
8   int main() {
9      const int SIZE = 20;
10     char input[SIZE];
11     int total = 0, count = 0;
12     double average;
13
14     cout << "To average a series of numbers.\n";
15     cout << "Enter the first number or Q to quit: ";
16     cin.getline(input, SIZE);
17     while (tolower(input[0]) != 'q') {
18        total += atoi(input);
19        count++;
20        cout << "Enter next number or Q to quit: ";
21        cin.getline(input, SIZE);
22     }
23     if (count != 0) {
24        average = static_cast<double>(total) / count;
25        cout << "Average: " << average << endl;
26     }
27     return 0;
28  }
```

Q#12

Given Program 6 which consists of a `void` function. Identify the output of the program. State the name of function, function header and function declaration in the program.

```
1   //Program 6
2   //void function example
3   #include <iostream>
4   using namespace std;
5
6   void printmessage();
7
8   int main() {
9     printmessage ();
10    return 0;
11  }
12
```

```
13  void printmessage () {
14    cout << "I'm a function!";
15  }
```

Q#13

Given Program 7 which consists of a function that passes data by value. Identify the output of the program. State the name of function, function header and function declaration in the program.

```
1   //Program 7
2   #include <iostream>
3   using namespace std;
4
5   int subtraction (int, int);
6
7   int main () {
8     int x = 5, y = 3, z;
9     z = subtraction (7,2);
10    cout << "The first result is " << z << '\n';
11    cout << "The second result is " << subtraction(7,2) << endl;
12    cout << "The third result is " << subtraction(x,y) << endl;
13    z = 4 + subtraction(x,y);
14    cout << "The fourth result is " << z << '\n';
15    return 0;
16  }
17
18  int subtraction (int a, int b) {
19    int r;
20    r = a - b;
21    return r;
22  }
```

Q#14

Given Program 8 which consists of a function that passes data by reference. Identify the output of the program. State the name of function, function header and function declaration in the program.

```
1   //Program 8
2   #include <iostream>
3   using namespace std;
4
5   void duplicate (int &a, int &b, int &c) {
6     a *= 2;
7     b *= 2;
8     c *= 2;
9   }
10
11  int main () {
```

```
12    int x = 1, y = 3, z = 7;
13    duplicate (x, y, z);
14    cout << "x = " << x << ", y = " << y << ", z = " << z;
15    return 0;
16  }
```

Q#15

Given Program 9 which consists of a function that is returning more than one value. Identify the output of the program. State the name of function, function header and function declaration in the program.

```
1   //Program 9
2   #include <iostream>
3   using namespace std;
4
5   void prevnext (int x, int &prev, int &next);
6
7   int main (){
8     int x = 100, y, z;
9     prevnext (x, y, z);
10    cout << "Previous = " << y << ", Next = " << z;
11    return 0;
12  }
13
14  void prevnext (int x, int &prev, int &next) {
15    prev = x - 1;
16    next = x + 1;
17  }
```

Q#15

Given Program 10 which consist of fixed value in a function. Identify the output of the program. State the name of function, function header and function declaration in the program.

```
1   //Program 10
2   #include <iostream>
3   using namespace std;
4
5   int divide (int a, int b = 2);
6
7   int main () {
8     cout << "20/default-value is: " << divide(20);
9     cout << endl;
10    cout << "20/4 is: " << divide(20, 4);
11    return 0;
12  }
13
14  int divide (int a, int b) {
15    int r;
```

```
16    r = a / b;
17    return r;
18  }
```

Q#16

Given Program 11 which consists of inline function. Identify the output of the program. State the name of function, function header and function declaration in the program.

```
1   //Program 11
2   #include <iostream>
3   using namespace std;
4
5   inline float cube(const float a) {return a * a * a;}
6
7   int main() {
8    float side;
9
10   cout << "Enter the side length of your cube: ";
11   cin >> side;
12   cout << "Volume of cube with side ";
13   cout << side << " is " << cube(side) << endl;
14
15   return 0;
16  }
```

Q#17

Given Program 12 which consists of overloaded functions. Identify the output of the program. State the name of function, function header and function declaration in the program.

```
1   //Program 12
2   #include <iostream>
3   using namespace std;
4
5   int calculateArea(int a, int b) {
6     return (a * b);
7   }
8
9   float calculateArea(float r) {
10    return (r * r * 3.142);
11  }
12
13  int main () {
14    int length = 5, width = 2;
15    float radius = 6.0;
16
17    cout << "\nThe area of a rectangle is: ";
18    cout << calculateArea(length,width);
19    cout << "\nThe area of a circle is:    ";
```

```
20    cout << calculateArea(radius);
21    cout << "\nEnd of program";
22
23    return 0;
24  }
```

Q#18

Given Program 13 that reads three integer numbers and find the maximum number among the three. Complete the program by writing a function named `maximum`, that finds the largest of the three numbers.

```
1   //Program 13
2   int maximum(int, int, int);
3
4   int main() {
5     int a, b, c ;
6
7     cout << "Enter Three Integers: ";
8     cin >> a >> b >> c;
9     cout << "Maximum is: " << maximum(a, b, c) << endl;
10
11    return 0;
12  }
```

Q#19

Given Program 14 that reads an integer number and determines whether the number is even or odd. Complete the program by writing a function named `is_even`, that prints "yes" if the number is even, or prints "no", otherwise.

```
1   //Program 14
2   #include <iostream>
3   using namespace std;
4
5   void is_even(int a); // Function Prototype
6
7   int main(void)
8   {
9     int a;
10    cout << "Enter an integer:\n";
11    cin >> a;
12    cout << "Is " << a << " even?";
13    is_even(a);
14    cout << endl;
15    return 0;
16  }
```

Q#20

Given Program 15 that reads the distance in miles and hours, calculates and prints miles per hour. Complete the program by writing the function prototype named `GetData` (at line 6), writing the code that invoke the function `GetData` (at line 16), and writing the body of the function `GetData` that reads miles and hours (at line 26).

```
1   //Program 15
2   #include <iostream>
3   #include <iomanip>
4   using namespace std;
5
6   /* Write the function prototype for GetData */
7
8   int main () {
9       float  miles;
10      float  hours;
11      float  milesPerHour;
12
13      cout << setprecision(5);
14
15  /* Write the code to invoke function GetData */
16
17      milesPerHour = miles / hours;
18      cout  << setw(10) << miles
19            << setw(10) << hours
20            << setw(10) << milesPerHour << endl;
21
22      return 0;
23  }
24
25  /* Write the body of the function GetData that read miles and hours */
```

Q#21

Given Program 16 that has an overloaded function named `average`, which finds the average of given numbers. Complete the program by writing the function prototype named `average` (at line 6 and 7) that sends 3 parameters, writing the code that invokes the function `average` (at line 17 and 18), and writing the body of the overloaded function `average` that finds the average of three numbers (at line 29).

```
1   //Program 16
2   #include <iostream>
3   using namespace std;
4
5   double average(int first_number, int second_number);
6   /* Write the function prototype for the overloaded function
7   average that send 3 parameters */
8
```

```
9   int main() {
10    int number_A = 5, number_B = 3, number_C = 10;
11
12    cout << "The integer average of " << number_A;
13    cout << << " and "<<number_B << " is ";
14    cout << average(number_A, number_B) << ".\n\n";
15
16  /* Write the code that invoke the function average with 3
17  parameters */
18
19  getch();
20  return 0;
21  }
22
23  double average(int first_number, int second_number) {
24    return ((first_number + second_number) / 2);
25  }
26
27  // write the body of the overloaded function average
```

Q#22

Dato' Vina has her own business and a founder of a few products. She has a few agents to sell her products. The agents will get some commission for each product that has been sold to the client. Write a complete C++ program that helps Dato' Vina to see the full sales report of her own products. The program should perform the following tasks:

Task 1: Write a function named ***readQty***.

a) This is a non-returning function.
b) It takes the quantity for Product A, Product B and Product C that have been sold by an agent as input parameters.
c) The function should ask the user to enter the quantity for each product. ***Input validation***: the quantity entered by user must be between 0 to 100 marks only. If the input is out of the range, the user must enter the quantity again. It sends all the values entered by the user in (c) back to the function caller / calling module using reference parameters.

Task 2: Write a function named ***displayHighProd***.

a) This is a non-returning function.
b) It takes the total of commission of each product for each agent as an input parameter.
c) The function should display the product that the agent got highest commission based on the percentage in Table 1.

Task 3: Write a function named ***calcAvrg.***

a) It takes the number of agents and total of commission for all agents as input parameters.

b) The function should calculate the average of commission for all agents.

c) It should return the average of commission calculated in (b).

Table 1

Product	Price (RM)	Percentage of Commission
A	150	2.5%
B	300	5%
C	450	10%

Task 4: Write a function named ***summaryCom***.

a) This is a non-returning function.
b) It takes the total of commission of each product and total of commission for all products for each agent as an input parameter.
c) The function should display the product that the agent got highest commission by calling the ***displayHighProd*** function.

Task 5: Write a main function to perform the following tasks:

a) You need to use an appropriate **LOOP** to perform the process in this function. The loop will be terminated when the user press ENTER without any value for student's name.
b) You are **NOT ALLOWED** to use **arrays** except an array of characters.
c) The function should ask the user to enter an agent's name.
d) The function may need to call the functions that are defined in Task 1, Task 3, and Task 4.
e) The function should calculate the total of commission of each product and total of commission for all products for each agent based on the following formula in Table 2.

Table 2

Total of commission for each product	Total Product A = quantity * ((150*0.025))
	Total Product B = quantity * ((300*0.05))
	Total Product C = quantity * ((450*0.1))
Total of commission for all products	Total Product A + Total Product B + Total Product C

f) The function should determine the agent with the highest and lowest total of commission for all products, and highest commission for each product, calculate the number of agents, and the average of commission for all agents. ***Note***: You are **NOT ALLOWED** to use any **predefined function(s)** to determine the highest and lowest total of commission.
g) The program should produce the output as in the sample execution given below. ***Note***: The values in **bold** are input by the user.

Task 6: List all function prototypes.

Task 7: You must ensure your program fulfil the following criteria:

a) The program can be run.
b) The program uses an appropriate structure for the program (e.g., all required header files are included, the program is properly written, proper indentation, etc.)

SAMPLE PROGRAM EXECUTION

```
----------------------AGENT-----------------------
Agent Name: AFIQ ROSDI
Product A: 56
Product B: 34
Product C: 20
-------------------SALES SUMMARY-------------------
Total of commission for three products: RM1620
Highest commission - Product C: RM900
----------------------AGENT-----------------------
Agent Name: SHIRA ATINI
Product A: 50
Product B: 87
Product C: 8
-------------------SALES SUMMARY-------------------
Total of commission for three products: RM1852.5
Highest commission - Product B: RM1305
----------------------AGENT-----------------------
Agent Name: ZAMRI HUSIN
Product A: 99
Product B: 21
Product C: 7
-------------------SALES SUMMARY-------------------
Total of commission for three products: RM1001.25
Highest commission - Product A: RM371.25
----------------------AGENT-----------------------
Agent Name: NANA ALIA
Product A: 30
Product B: 100
Product C: 110
Product C: -1
Product C: 11
-------------------SALES SUMMARY-------------------
Total of commission for three products: RM2107.5
Highest commission - Product B: RM1500
----------------------AGENT-----------------------
Agent Name:
------------------SALES ANALYSIS-------------------
Lowest commission for three products: RM 1001.25 (ZAMRI HUSIN)
Highest commission for three products: RM 2107.5 (NANA ALIA)
Highest commission for Product A: RM 371.25 (ZAMRI HUSIN)
Highest commission for Product B: RM 1500 (NANA ALIA)
Highest commission for Product C: RM 900 (AFIQ ROSDI)
Total of commission for 4 agents: RM 6581.25
Average commission for 4 agents: RM 1645.31
```

Q#23

You are given a C++ program (**`Program 1`**) 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   //Program 1
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

Your Covid-19 risk status is GREEN
```

Sample Output for Program Execution 2

```
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): y

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

Your Covid-19 risk status is ORANGE
```

Sample Output for Program Execution 3

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

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): n

Your Covid-19 risk status is YELLOW
```

Sample Output for Program Execution 4

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

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

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

Your Covid-19 risk status is RED
```

Figure 1: The example of outputs

Q#24

You are given a C++ program (**`Program 2`**) 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 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 2**. ***Note:*** The values in **bold** are input by the user.

```
1   //Program 2
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 2: The example of outputs