(SECJ1013) PROGRAMMING TECHNIQUE 1

SEM 1, SESSION 2023/2024

LAB EXERCISE 3 (4%)

INSTRUCTIONS TO THE STUDENTS

- This exercise must be done **individually**.
- Any form of plagiarism is **NOT ALLOWED**. Students who copied other students' assignments will get **ZERO** marks (both parties, students who copied, and students who shared their work).

SUBMISSION PROCEDURE

- Please submit this exercise no later than **December 21, 2023, Thursday (1 PM MYT)**.
- Only hardcopy is accepted for this submission with handwriting (at my office – n28a, level 2, room 02-31-01).

Fill your information here:

Name	**Evelyn Goh Yuan Qi**
Matric Number	**A23CS0222**
Year / Course	**2023/SECPH**
Section	**3**

SET 1

The following program code has errors. Locate the errors.

Line	C++ Codes
1	`#include <iostream>`
2	`using namespace std;`
3	
4	`// function prototypes`
5	`void display_Question()`
6	`void yes_No()`
7	`int get_Status()`
8	
9	`// start main function`
10	`int main() {`
11	`    char red_zone, close_contact, fever;`
12	`     // two possible character values only:`
13	`     // 'y' -> yes, 'n' -> no`
14	
15	`    int status;`
16	`     // 0 -> GREEN, 1 -> YELLOW, 2 -> ORANGE, 3 -> RED`
17	
18	`    for (int i = 1; i < 3; i++) {`
19	`        display_question(i);`
20	`        if (i = 1)`
21	`            yes_No(red_zone);`
22	`               // set red_zone either 'y' or 'n'`
23	`        else if (i = 2)`
24	`            yes_No(close_contact);`
25	`               // set close_contact either 'y' or 'n'`
26	`        else`
27	`            yes_No(fever);`
28	`               // set fever either 'y' or 'n'`
29	`    }`
30	
31	

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

```
62          cin << ans;
63      } while (ans = 'n' & ans = 'y');
64      cout << '\n';
65  }
66
67  int get_Status(char rz, char cc, char f) {
68      int s = 0;
69      if (rz = 'y') s++;
70      if (cc = 'y') s++;
71      if (f = 'y') s++;
72      return s;
    }
```

Fill in the following table by stating the line number and write the correct statement with the reason(s).

Line Number	**Correct Statement with the reason(s)**
5	void display_Question(int); (must end programming statement with semicolon and write data type for parameter)
6	void yes_No(char&); (must end programming statement with semicolon, write data type for parameter and add ampersand for reference variable)
7	int get_Status(char,cher,char); (must end programming statement with semicolon and write data type for parameter)
18	for (int i = 1; i <= 3; i++) { (< change to <= because have 3 questions)
19	display_Question(i); (name for function call must same with function prototype)
20	if (i == 1) (must use relational operator “==”)
23	else if (i == 2) (must use relational operator “==”)
35	switch (status) { (use switch statement instead of while)
36	case 0: cout << "GREEN"; break; (use case)
37	case 1: cout << "YELLOW"; break; (use case)
38	case 2: cout << "ORANGE"; break; (use case and add break)
39	case 3: cout << "RED"; break; (use case and add break)
46	void display_Question(int q) { (number is int not char)
55	Celcius?\n"; break; (add break for switch case)
56	} (remove return q because void is non returning data type)
59	void yes_No(char& ans) { (Add ampersand for reference variable)
62	cin >> ans; (symbol for cin is >>)
63	} while (ans != 'n' && ans != 'y'); (use logical and relational operator)
69	if (rz == 'y') s++; (must use relational operator “==”)
70	if (cc == 'y') s++; (must use relational operator “==”)
71	if (f == 'y') s++; (must use relational operator “==”)

Line Number	Correct Statement with the reason(s)

Line Number	Correct Statement with the reason(s)

SET 2

Complete the code segments in the program below.

1. Task 1:

   Write a function named **`setType`** with the parameter of car type variable,
   which prompts the user to enter a car type either "**`sedan`**" or "**`mpv`**"
   and continues to do so in a loop until the entered type is either "**`sedan`**" or "**`mpv`**".

2. Task 2:

   Write a function named **`setPackage`** with the parameter of car wash service package,
   which displays a menu with three options for car wash service packages: **`Basic (1)`**, **`Deluxe (2)`**, and **`Premium (3)`**.
   It prompts the user to choose a package by entering the corresponding number (**`1`**, **`2`**, or **`3`**).
   The loop continues until a valid package number (between **`1`** and **`3`** inclusive) is entered by the user.

3. Task 3:

   Write a function named **`wash`** with the parameter of car type variable,
   which calculates the wash service charge based on the type of car.
   If the car type is "**`mpv`**", the charge is calculated as **1.2 times** the constant **`WASH`**;
   otherwise, the charge is equal to the constant **`WASH`**.
   The calculated charge is then displayed, and the function returns the computed charge.

4. Task 4:

   Write a function named **`vacuum`** with the parameter of car type variable,
   which calculates the vacuum service charge based on the type of car.
   If the car type is "**`mpv`**", the charge is calculated as **1.05 times** the constant **`VACUUM`**;
   otherwise, the charge is equal to the constant **`VACUUM`**.
   The calculated charge is then displayed, and the function returns the computed charge.

5. Task 5:

   Write a function named **`polish`** with the parameter of car type variable,
   which calculates the polish service charge based on the type of car.
   If the car type is "**`mpv`**", the charge is calculated as **1.2 times** the constant **`POLISH`**;
   otherwise, the charge is equal to the constant **`POLISH`**.
   The calculated charge is then displayed, and the function returns the computed charge.

6. Task 6:

   List all function prototypes.

7. Task 7:

   (i) Call the functions from Task 1 until Task 5 in the **`main`** function.

   (ii) **`totalCharge`** is the variable to hold the total service charge based on different wash service package with different car type ("**`sedan`**" or "**`mpv`**").

   (Note for wash service packages: Task 3 for **`Basic`**, Task 4 for **`Deluxe`**, and Task 5 for **`Premium`**)

   (iii) Print out the final total service charge.

8. Task 8:

   You must ensure your program fulfil the following criteria:

   - The program is able to run.
   - 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 Execution Output

```
Enter car type (sedan/mpv): sedan

1. Basic
2. Deluxe
3. Premium
Choose wash service package (1/2/3): 1

Wash service charge is 10

Total service charge is 10
--------------------------------
```

```
Enter car type (sedan/mpv): sedan

1. Basic
2. Deluxe
3. Premium
Choose wash service package (1/2/3): 2

Vacuum service charge is 7

Total service charge is 7
--------------------------------
```

```
Enter car type (sedan/mpv): sedan

1. Basic
2. Deluxe
3. Premium
Choose wash service package (1/2/3): 3

Polish service charge is 15

Total service charge is 15
--------------------------------
```

```
Enter car type (sedan/mpv): mpv

1. Basic
2. Deluxe
3. Premium
Choose wash service package (1/2/3): 1

Wash service charge is 12

Total service charge is 12
--------------------------------
```

```
Enter car type (sedan/mpv): mpv

1. Basic
2. Deluxe
3. Premium
Choose wash service package (1/2/3): 2

Vacuum service charge is 7.35

Total service charge is 7.35
--------------------------------
```

```
Enter car type (sedan/mpv): mpv

1. Basic
2. Deluxe
3. Premium
Choose wash service package (1/2/3): 3

Polish service charge is 18

Total service charge is 18
--------------------------------
```

Note: ▬ show user's input.

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

// constants with the associated values
#define WASH 10.0 // the price of WASH service charge
#define VACUUM 7.0 // the price of VACUUM service charge
#define POLISH 15.0 // the price of POLISH service charge

// Task 6: List all function prototypes.
void setType(string &type);
void setPackage(int &pkg);
float wash(string type);
float vacuum(string type);
float polish(string type);


// Task 7: (i) Call the functions from Task 1 until Task 5.
int main() {
     string carType; // car type variable
     int wsPkg; // car wash service package
     float totalCharge = 0; // total service charge based on
different wash service package with different car type

     // call setType function with the parameter carType
     setType(carType);

     // call setPackage function with the parameter wsPkg
     setPackage(wsPkg);


     // Task 7: (ii) totalCharge is the variable to hold the
total service charge based on different wash service package
with different car type (sedan or mpv).
```

```
      switch (wsPkg) {
      case 1: totalCharge += wash(carType);
              break;
      case 2: totalCharge += vacuum(carType);
             break;
      case 3: totalCharge += polish(carType);
             break;
      default: cout << "Invalid wash service package choice."
<< endl;
      }
      cout << endl;
      cout << "Total service charge is " << totalCharge;
      return 0;
}

// Task 1: Function to set car type
void setType(string &type) {
do {
        cout << "Enter car type (sedan/mpv): ";
        cin >> type;
    } while (type != "sedan" && type != "mpv");

      cout << endl;
}

// Task 2: Function to set wash service package
void setPackage(int &pkg) {
do {
        cout << "1. Basic\n2. Deluxe\n3. Premium\n";
        cout << "Choose wash service package (1/2/3): ";
        cin >> pkg;
    } while (pkg < 1 || pkg > 3);
```

```
     cout << endl;
}

// Task 3: Function to determine exterior wash service charge
based on car type
float wash(string type) {
float charge;
if(type == "mpv")
   charge = 1.2 * WASH;
else
   charge=WASH;
cout << "Wash service charge is " << charge << endl;
return charge;
}

// Task 4: Function to determine interior vacuum service
charge based on car type
float vacuum(string type) {
float charge;
if(type == "mpv")
     charge = 1.05 * VACUUM;
else
      charge = VACUUM;
cout << "Vacuum service charge is " << charge << endl;
return charge;
}

// Task 5: Function to determine exterior polish service
charge based on car type
float polish(string type) {
float charge;
if(type == "mpv")
    charge = 1.2 * POLISH;
else
```

```
    charge = POLISH;
cout << "Polish service charge is " << charge << endl;
return charge;
}
```