

Set 1

Line number	Correction Statement with the reason(s)
5	void display-Question(int); // function prototype need to declare the data type for variable in bracket and need a semicolon
6	void yes-No(char k); // function prototype need to declare the data type for variable in bracket and need a semicolon. Also, the parameter must be a reference variable to pass the value of ans to main function.
7	int get_Status(char, char, char) // function prototype need to declare the data type for variable in the bracket and need a semicolon.
18	for(int i=1; i<=3; i++) { // continue loop if (i==3), to display the question "Body temperature >= 38 degrees Celcius?\n"; The loop needs to run 3 times.
19	display-Question(i); // display-question(i) is not same with display-Question(i)
20	if(i==1) // We need to check whether i is equal to 1 or not, instead of using single equal symbol to assign 1 to the variable i.
23	Else if(i==2) // We need to check whether i is equal to 2 or not, instead of using single symbol to assign 2 to the variable i.
35	switch(status) { // It is not a loop because while loop will lead to infinite loop
36	case 0: cout << "GREEN"; break; // change to switch condition
37	case 1: cout << "YELLOW"; break;
38	case 2: cout << "ORANGE"; break; // break out of the switch
39	default: cout << "RED"; break;
46	void display-Question(int q) { // the parameter q is a integer
47	switch(q) { // need a open bracket for switch statements
54	default: // We usually use default statements for last line of the switch statement
55	Celcius?\n"; break; // break out of the switch
56	} // void function cannot have return, not a return function
59	void yes-No(char &ans) { // need the ampersand (&) symbol to indicate that the parameter is a reference variable so that can pass the value by reference.

Line number	Correction statement with the reason(s)
62	<code>cin >> ans ;</code> // input use >> symbol
63	<code>} while (ans != 'n' && ans != 'y') ;</code> // the loop continues until a valid answer which is y or n is entered by the user
64	<code>cout << "\n" ;</code> // string is use double quote symbol
69 70 71	<code>if (rz == 'y') s++ ;</code> // single equal symbol is used to assign a value to a variable, <code>if (cc == 'y') s++ ;</code> but here, we need to check and compare the values. <code>if (f == 'y') s++ ;</code> For example, we are checking if rz is equal to y or not instead of assign y to rz.

SET 2

```
#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&);
```

```
void setPackage (int&);
```

```
float wash (string);
```

```
float vacuum (string);
```

```
float polish (string);
```

```
// 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
```

```
    // 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;
```

```
    default: totalCharge += polish (carType); break;
```

```
}
```



```

    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) {
    cout << "1. Basic" << endl;
    cout << "2. Deluxe" << endl;
    cout << "3. Premium" << endl;
}

```

```

do
{
    cout << "Choose wash service package (1/2/3): ";
    cin >> pkg;
} while (pkg != 1 && pkg != 2 && 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;
if(type == "sedan")
    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;
    if(type == "sedan")
        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;
    if(type == "sedan")
        charge = POLISH;

    cout << "Polish service charge is " << charge << endl;
    return charge;
}

```