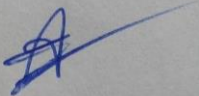


Hei Wang Andre Law

4017 5600

Assignment 4

I certify that this submission is my original work and
meets the faculty's Expectations of Originality.

4017 5600 

Question 1

```
1 // Assignment 4, Question 1, Week 10
2 // Hei Wang Andre Law, 4017 5600
3 // Program: Multiplication of Row Element of Two Dimensional Array
4
5 #include<iostream>
6 #include<array>
7 using namespace std;
8
9 // declare array size and function prototype for the multiplication
10 const size_t cols{ 6 };
11 const size_t rows{ 5 };
12 void mult(array<array<int, cols>, rows>);
13
14 int main() {
15     array<array<int, cols>, rows> arrayMult; // declare the array
16     int num; // declare the integer input
17
18     cout << "Fill in the two-dimensional array A[5][6].\n\n";
19
20     // for-loop filling in the array elements from user inputs
21     for (size_t i{ 0 }; i < rows; i++) { // loop 5 times
22         for (size_t j{ 0 }; j < cols; j++) { // loop 6 times
23             cin >> num; // ask user a number
24             arrayMult[i][j] = num; // store that number in corresponding element
25         }
26     }
27     mult(arrayMult); // 'mult' function call
28     return 0;
29 }
30
31 // 'mult' function printing the multiplication of each row
32 void mult(array<array<int, cols>, rows> arrayMult){
33     // for-loop multiplying elements of each row
34     for (size_t i{ 0 }; i < rows; i++) { // loop 5 times
35         int result = 1; // set initial result to 1
36         for (size_t j{ 0 }; j < cols; j++) { // loop 6 times
37             // multiply 'result' by array element in [i][j] position
38             result = result * arrayMult[i][j];
39         }
40     }
41 }
```

Microsoft Visual Studio Debug Console

Fill in the two-dimensional array A[5][6].

```
2 3 4 6 2 1
3 1 0 4 5 8
3 1 0 4 5 2
3 1 1 4 5 9
3 1 3 4 1 8
```

The multiplication of row 0 is 288
The multiplication of row 1 is 0
The multiplication of row 2 is 0
The multiplication of row 3 is 540
The multiplication of row 4 is 288

E:\vsCode\Project Location\COEN 243 Assignment 0.
To automatically close the console when debugging stops.
Press any key to close this window . . .

Question 1

```
question1 (Global Scope)
25     }
26     }
27     mult(arrayMult); // 'mult' function call
28     return 0;
29 }
30
31 // 'mult' function printing the multiplication of each row
32 void mult(array<array<int, cols>, rows> arrayMult){
33     // for-loop multiplying elements of each row
34     for (size_t i{ 0 }; i < rows; i++) { // loop 5 times
35         int result = 1; // set initial result to 1
36         for (size_t j{ 0 }; j < cols; j++) { // loop 6 times
37             // multiply 'result' by array element in [i][j] position
38             result = result * arrayMult[i][j];
39         }
40         // print final result for each row
41         cout << "\nThe multiplication of row " << i << " is " << result;
42     }
43     cout << endl;
44 }
```

Microsoft Visual Studio Debug Console

Fill in the two-dimensional array A[5][6].

```
2 3 4 6 2 1
3 1 0 4 5 8
3 1 0 4 5 2
3 1 1 4 5 9
3 1 3 4 1 8
```

The multiplication of row 0 is 288
The multiplication of row 1 is 0
The multiplication of row 2 is 0
The multiplication of row 3 is 540
The multiplication of row 4 is 288

E:\vsCode\Project Location\COEN 243 Assignments\Assignment 4\question1\Debug\question1.exe (process 7404)
de 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically
le when debugging stops.
Press any key to close this window . . .

121 % No issues found

Question 2, testdevice.cpp

```
1 // Assignment 4, Question 2, Week 10
2 // Hei Wang Andre Law, 4017 5600
3 // Program: user-defined class housing info of electronic devices
4 // (cpp file of main function)
5
6 #include<iostream>
7 #include <string>
8 #include "device.h"
9 using namespace std;
10
11 int main() {
12     Edevice e1; // instantiate object e1 of class Edevice
13     Edevice e2; // instantiate object e2 of class Edevice
14     // declare the variables
15     string cat; // 'category'
16     int num; // 'model number'
17     string col; // 'color'
18     bool stat; // 'status'
19     int year; // 'year'
20     double price; // 'price'
21
22     int checkNum = 0;
23     int editNum = 0;
24
25     // for-loop twice for each Edevice
26     for (int i = 1; i <= 2; i++) {
27         // ask user to enter information of each attributes
28         cout << "\nEnter the information for device number " << i;
29         cout << "\n\nCategory: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch') ";
30         cin >> cat; // 'category' input
31         // while-loop until input corresponds to one of the option
32         while (cat != "Smartphone" && cat != "Tablet" && cat != "Laptop" && cat != "Smartwatch") {
33             cout << "Please try again...\nCategory: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch')";
34             cin >> cat;
35         }
36         cout << "Model Number: ";
37         cin >> num; // 'model nuber' input
38         cout << "Color: ";
```

Microsoft Visual Studio Debug Console

Enter the information for device number 1

Category: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch') Smartphone
Model Number: 1200
Color: red
Status: (0: false, 1:true) 0
Year Built: 2010
Price: 500

Enter the information for device number 2

Category: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch') Laptop
Model Number: 5700
Color: blue
Status: (0: false, 1:true) 1
Year Built: 2020
Price: 1000

Enter a number to perform one of the following commands:

1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of edevice Two
5. Exit Program

1

The attributes for this device are:

Category: Smartphone
Model Number: 1200
Color: red

Question 2, testdevice.cpp

```
Eddevice.cpp  device.h  testdevice.cpp # X
question2 (Global Scope)

34         cin >> cat;
35     }
36     cout << "Model Number: ";
37     cin >> num; // 'model nuber' input
38     cout << "Color: ";
39     cin >> col; // 'color' input
40     cout << "Status: (0: false, 1:true) ";
41     cin >> stat; // 'status' input
42     cout << "Year Built: ";
43     cin >> year; // 'year' input
44     cout << "Price: ";
45     cin >> price; // 'price' input
46
47     // first loop means i is 1, thus storing the data in e1
48     if (i == 1) {
49         e1.category(cat); // set category of the device
50         e1.color(col); // set the color of the device
51         e1.modelNumber(num); // set the model number of the device
52         e1.status(stat); // set the status of the device
53         e1.yearBuilt(year); // set the year of the device
54         e1.priceD(price); // set the price of the device
55     }
56     // second loop means i is 2, thus storing the data in e2
57     else {
58         e2.category(cat); // set the category of the device
59         e2.color(col); // set the color of the device
60         e2.modelNumber(num); // set the model number of the device
61         e2.status(stat); // set the status of the device
62         e2.yearBuilt(year); // set the year of the device
63         e2.priceD(price); // set the price of the device
64     }
65 }
66 // loop as long as user doesn't input '5'
67 // this loop allows for user to interact with the member functions
68 while (checkNum != 5) {
69     cout << "\nEnter a number to perform one of the following commands:";
70     cout << "\n1. Display the Information of Edevice One";
71     cout << "\n2. Display the Informtion of Edevice Two";

```

Select Microsoft Visual Studio Debug Console

The attributes for this device are:

Category: Smartphone
Model Number: 1200
Color: red
Status: 0
Year Built: 2010
Price: 500

Enter a number to perform one of the following commands:

1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of edevice Two
5. Exit Program

2

The attributes for this device are:

Category: Laptop
Model Number: 5700
Color: blue
Status: 1
Year Built: 2020
Price: 1000

Enter a number to perform one of the following commands:

1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of Edevice Two
5. Exit Program

4

Enter a number corresponding an attribute to edit

1. Category
2. Model number
3. Color
4. Status
5. Year Built
6. Price

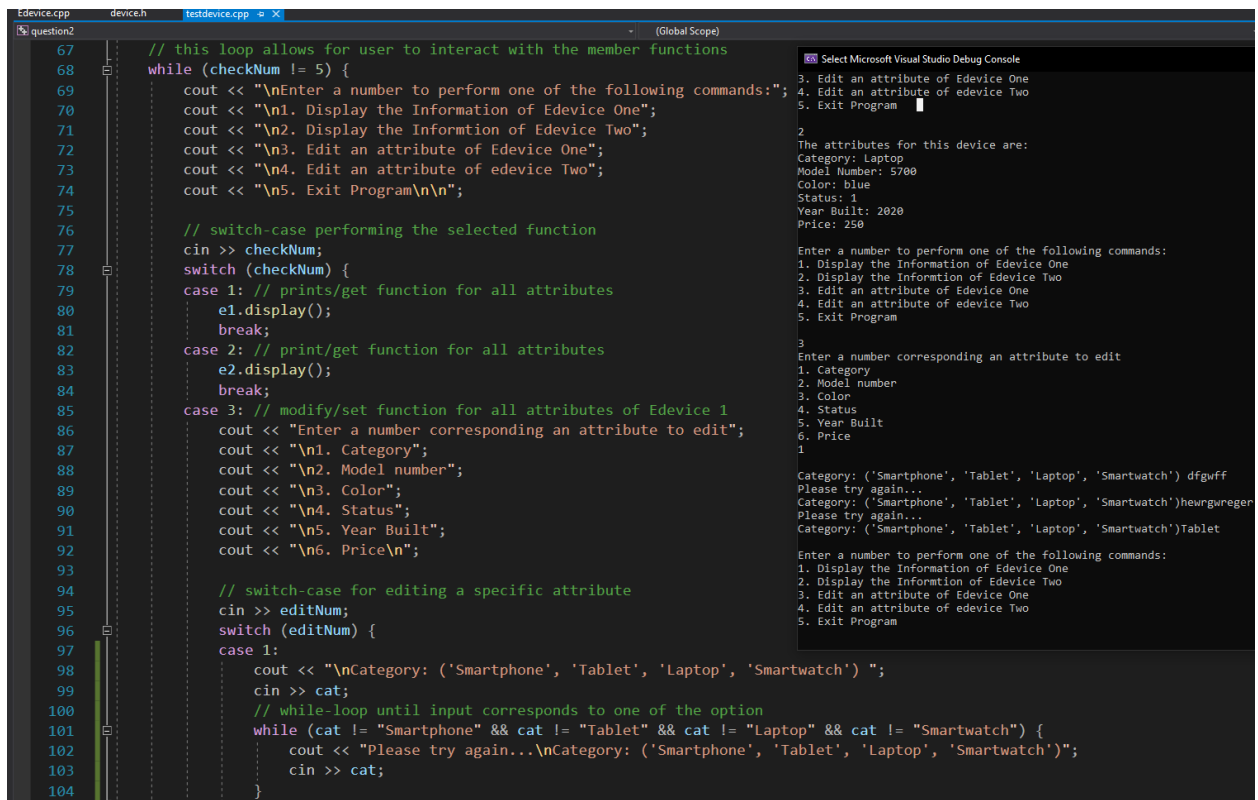
6

Price: 250

Enter a number to perform one of the following commands:

1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of edevice Two
5. Exit Program

Question 2, testdevice.cpp



```
67 // this loop allows for user to interact with the member functions
68 while (checkNum != 5) {
69     cout << "\nEnter a number to perform one of the following commands:";
70     cout << "\n1. Display the Information of Edevice One";
71     cout << "\n2. Display the Information of Edevice Two";
72     cout << "\n3. Edit an attribute of Edevice One";
73     cout << "\n4. Edit an attribute of Edevice Two";
74     cout << "\n5. Exit Program\n\n";
75
76     // switch-case performing the selected function
77     cin >> checkNum;
78     switch (checkNum) {
79     case 1: // prints/get function for all attributes
80         e1.display();
81         break;
82     case 2: // print/get function for all attributes
83         e2.display();
84         break;
85     case 3: // modify/set function for all attributes of Edevice 1
86         cout << "Enter a number corresponding an attribute to edit";
87         cout << "\n1. Category";
88         cout << "\n2. Model number";
89         cout << "\n3. Color";
90         cout << "\n4. Status";
91         cout << "\n5. Year Built";
92         cout << "\n6. Price\n";
93
94         // switch-case for editing a specific attribute
95         cin >> editNum;
96         switch (editNum) {
97         case 1:
98             cout << "\nCategory: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch') ";
99             cin >> cat;
100             // while-loop until input corresponds to one of the option
101             while (cat != "Smartphone" && cat != "Tablet" && cat != "Laptop" && cat != "Smartwatch") {
102                 cout << "Please try again...\nCategory: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch')";
103                 cin >> cat;
104             }
```

Select Microsoft Visual Studio Debug Console

```
3. Edit an attribute of Edevice One
4. Edit an attribute of Edevice Two
5. Exit Program
1
The attributes for this device are:
Category: Laptop
Model Number: 5700
Color: blue
Status: 1
Year Built: 2020
Price: 250
Enter a number to perform one of the following commands:
1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of Edevice Two
5. Exit Program
3
Enter a number corresponding an attribute to edit
1. Category
2. Model number
3. Color
4. Status
5. Year Built
6. Price
1
Category: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch') dfgvff
Please try again...
Category: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch')hewngwreger
Please try again...
Category: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch')Tablet
Enter a number to perform one of the following commands:
1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of Edevice Two
5. Exit Program
```

Question 2, testdevice.cpp

```
testdevice.cpp  device.h  testdevice.cpp  X
question2  (Global Scope)

100 // while-loop until input corresponds to one of the option
101 while (cat != "Smartphone" && cat != "Tablet" && cat != "Laptop" && cat != "Smartwatch") {
102     cout << "Please try again...\nCategory: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch')";
103     cin >> cat;
104 }
105 e1.category(cat); // set category of the device
106 break;
107 case 2:
108     cout << "Model Number: ";
109     cin >> num;
110     e1.modelNumber(num); // set the model number of the device
111     break;
112 case 3:
113     cout << "Color: ";
114     cin >> col;
115     e1.color(col); // set the color of the device
116     break;
117 case 4:
118     cout << "Status: (0: false, 1:true): ";
119     cin >> stat;
120     e1.status(stat); // set the status of the device
121     break;
122 case 5:
123     cout << "Year Built: ";
124     cin >> year;
125     e1.yearBuilt(year); // set the year of the device
126     break;
127 case 6:
128     cout << "Price: ";
129     cin >> price;
130     e1.priceD(price); // set the year of the device
131     break;
132 }
133 break;
134 case 4: // modify/set function for all attributes of Edevice 2
135     cout << "Enter a number corresponding an attribute to edit";
136     cout << "\n1. Category";
137     cout << "\n2. Model number";
```

Select Microsoft Visual Studio Debug Console
Enter a number to perform one of the following commands:
1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of edevice Two
5. Exit Program
1
The attributes for this device are:
Category: Tablet
Model Number: 1200
Color: red
Status: 0
Year Built: 2010
Price: 500
Enter a number to perform one of the following commands:
1. Display the Information of Edevice One
2. Display the Information of Edevice Two
3. Edit an attribute of Edevice One
4. Edit an attribute of edevice Two
5. Exit Program
5
You have exited the program...
E:\vsCode\Project Location\COEN 243 Assignments\Assignment 4
de 0.
To automatically close the console when debugging stops, ena
le when debugging stops.
Press any key to close this window . . .

121 % No issues found

Question 2, testdevice.cpp

```
Edevice.cpp  device.h  testdevice.cpp x
question2  (Global Scope)

136     cout << "\n1. Category";
137     cout << "\n2. Model number";
138     cout << "\n3. Color";
139     cout << "\n4. Status";
140     cout << "\n5. Year Built";
141     cout << "\n6. Price\n";
142
143     // switch-case for editing a specific attribute
144     cin >> editNum;
145     switch (editNum) {
146     case 1:
147         cin >> cat;
148         // while-loop until input corresponds to one of the option
149         while (cat != "Smartphone" && cat != "Tablet" && cat != "Laptop" && cat != "Smartwatch") {
150             cout << "Please try again...\nCategory: ('Smartphone', 'Tablet', 'Laptop', 'Smartwatch')";
151             cin >> cat;
152         }
153         e2.category(cat); // set category of the device
154         break;
155     case 2:
156         cout << "Model Number: ";
157         cin >> num;
158         e2.modelNumber(num); // set the model number of the device
159         break;
160     case 3:
161         cout << "Color: ";
162         cin >> col;
163         e2.color(col); // set the color of the device
164         break;
165     case 4:
166         cout << "Status: (0: false, 1:true): ";
167         cin >> stat;
168         e2.status(stat); // set the status of the device
169         break;
170     case 5:
171         cout << "Year Built: ";
172         cin >> year;
173         e2.yearBuilt(year); // set the year of the device
```

121 % No issues found

Question 2, testdevice.cpp

```
Edevice.cpp  device.h  testdevice.cpp  X
question2  (Global Sco

163         e2.color(col); // set the color of the device
164         break;
165     case 4:
166         cout << "Status: (0: false, 1:true): ";
167         cin >> stat;
168         e2.status(stat); // set the status of the device
169         break;
170     case 5:
171         cout << "Year Built: ";
172         cin >> year;
173         e2.yearBuilt(year); // set the year of the device
174         break;
175     case 6:
176         cout << "Price: ";
177         cin >> price;
178         e2.priceD(price); // set the year of the device
179         break;
180     }
181     break;
182     case 5: // exits program
183         cout << "You have exited the program...";
184         break;
185     }
186 }
187 return 0;
188 }
```

Question 2, device.h

```
Edevice.cpp  device.h  testdevice.cpp
question2  Edevice

1  // Assignment 4, Question 2, Week 10
2  // Hei Wang Andre Law, 4017 5600
3  // Program: user-defined class housing info of electronic devices
4  // (header file containing the specification of the class)
5
6  #include <string>
7  using namespace std;
8
9  // class definition for Edevice
10 class Edevice {
11     private: // data members
12         string cat; // category of the Edevice
13         int mod; // model number of the Edevice
14         string col; // color of the Edevice
15         bool stat; // status of the Edevice
16         int year; // year of the Edevice
17         double price; // price of the Edevice
18
19     public: // member function
20         Edevice(); // constructor
21         void category(string cat); // set category of the device
22         void modelNumber(int); // set model number of the device
23         void color(string col); // set color of the device
24         void status(bool stat); // set status of the device
25         void yearBuilt(int year); // set year of the device
26         void priceD(double price); // set price of the device
27         void display(); // print/get all attributes
28     }; // end of class Edevice
```

Question 2, Edevice.cpp

```
Edevice.cpp  x device.h  testdevice.cpp
question2 (Global Sc

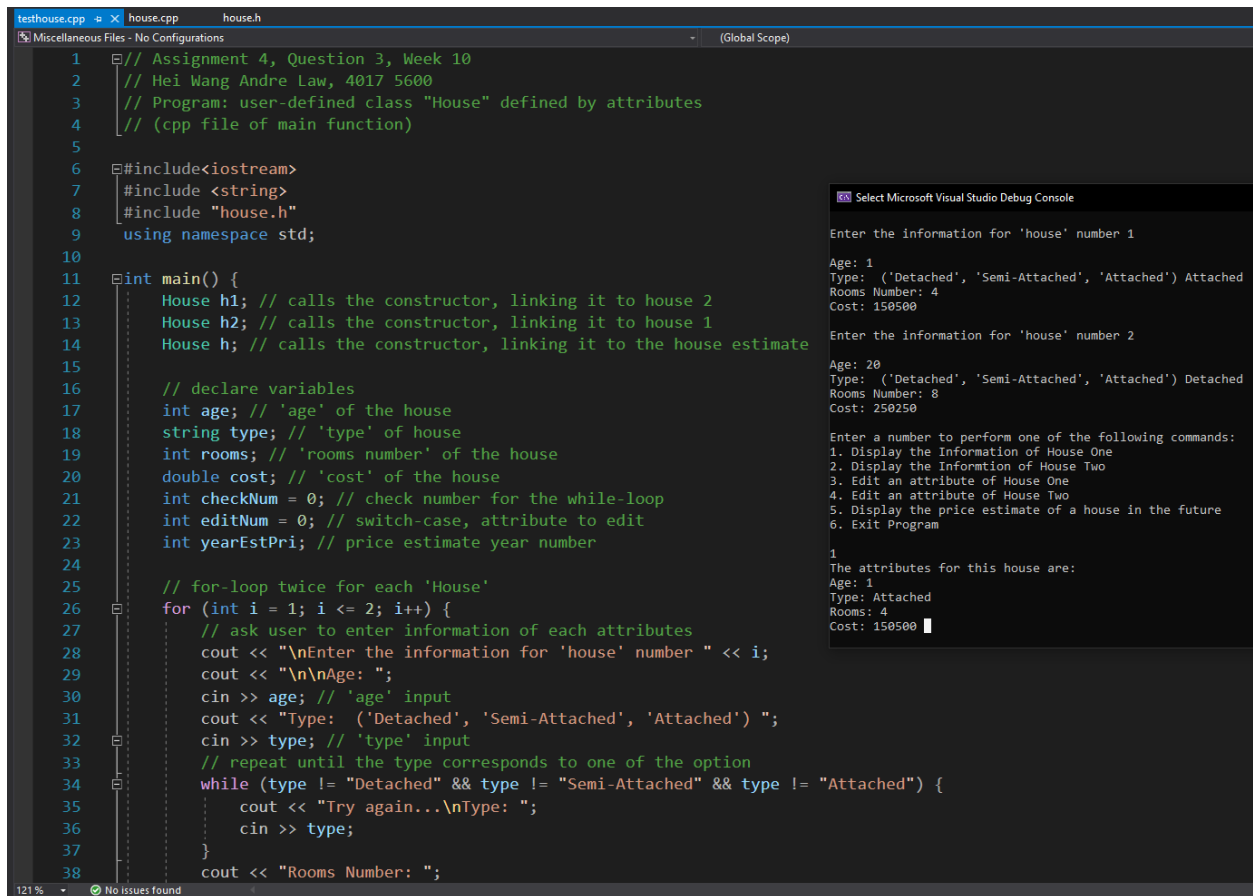
1  // Assignment 4, Question 2, Week 10
2  // Hei Wang Andre Law, 4017 5600
3  // Program: user-defined class housing info of electronic devices
4  // (cpp file for implementing the member functions of the class)
5
6  #include <iostream>
7  #include <string>
8  #include "device.h"
9  using namespace std;
10
11  // class implementation of Edevice
12  Edevice::Edevice() {
13      // declare the variables for each attributes
14      cat = "";
15      mod = 0;
16      col = "";
17      stat = true;
18      year = 0;
19      price = 0.0;
20  }
21  // 'category' attribute
22  void Edevice::category(string category) {
23      cat = category;
24  }
25  // 'model number' attribute
26  void Edevice::modelNumber(int number) {
27      mod = number;
28  }
29  // 'color' attribute
30  void Edevice::color(string color) {
31      col = color;
32  }
33  // 'status' attribute
34  void Edevice::status(bool status) {
35      stat = status;
36  }
37  // 'year' attribute
38  void Edevice::yearBuilt(int yearBuilt) {
```

121 % No issues found

Question 2, Edevice.cpp

```
Edevice.cpp  X  device.h  testdevice.cpp
question2
31     col = color;
32 }
33 // 'status' attribute
34 void Edevice::status(bool status) {
35     stat = status;
36 }
37 // 'year' attribute
38 void Edevice::yearBuilt(int yearBuilt) {
39     year = yearBuilt;
40 }
41 // 'price' attribute
42 void Edevice::priceD(double pri) {
43     price = pri;
44 }
45 // print all attributes
46 void Edevice::display() {
47     cout << "The attributes for this device are:";
48     cout << "\nCategory: " << cat;
49     cout << "\nModel Number: " << mod;
50     cout << "\nColor: " << col;
51     cout << "\nStatus: " << stat;
52     cout << "\nYear Built: " << year;
53     cout << "\nPrice: " << price;
54     cout << endl;
55 }
```

Question 3, testhouse.cpp



The screenshot displays the Visual Studio IDE with the `testhouse.cpp` file open. The code is a C++ program for a house management system. It includes headers for `iostream`, `string`, and `house.h`, and uses the `std` namespace. The `main` function contains a loop that prompts the user to enter information for two houses. For each house, it asks for age, type (Detached, Semi-Attached, Attached), rooms number, and cost. It also includes a menu for displaying house information, editing attributes, or estimating future prices.

```
1 // Assignment 4, Question 3, Week 10
2 // Hei Wang Andre Law, 4017 5600
3 // Program: user-defined class "House" defined by attributes
4 // (cpp file of main function)
5
6 #include<iostream>
7 #include <string>
8 #include "house.h"
9 using namespace std;
10
11 int main() {
12     House h1; // calls the constructor, linking it to house 2
13     House h2; // calls the constructor, linking it to house 1
14     House h; // calls the constructor, linking it to the house estimate
15
16     // declare variables
17     int age; // 'age' of the house
18     string type; // 'type' of house
19     int rooms; // 'rooms number' of the house
20     double cost; // 'cost' of the house
21     int checkNum = 0; // check number for the while-loop
22     int editNum = 0; // switch-case, attribute to edit
23     int yearEstPri; // price estimate year number
24
25     // for-loop twice for each 'House'
26     for (int i = 1; i <= 2; i++) {
27         // ask user to enter information of each attributes
28         cout << "\nEnter the information for 'house' number " << i;
29         cout << "\n\nAge: ";
30         cin >> age; // 'age' input
31         cout << "Type: ('Detached', 'Semi-Attached', 'Attached') ";
32         cin >> type; // 'type' input
33         // repeat until the type corresponds to one of the option
34         while (type != "Detached" && type != "Semi-Attached" && type != "Attached") {
35             cout << "Try again...\nType: ";
36             cin >> type;
37         }
38         cout << "Rooms Number: ";
```

The Debug Console on the right shows the program's execution. It prompts for house information and displays the entered values for two houses. The first house has an age of 1, is 'Attached', has 4 rooms, and costs 150500. The second house has an age of 20, is 'Detached', has 8 rooms, and costs 250250. The console also shows a menu of commands and the attributes for the first house.

```
Select Microsoft Visual Studio Debug Console
Enter the information for 'house' number 1
Age: 1
Type: ('Detached', 'Semi-Attached', 'Attached') Attached
Rooms Number: 4
Cost: 150500
Enter the information for 'house' number 2
Age: 20
Type: ('Detached', 'Semi-Attached', 'Attached') Detached
Rooms Number: 8
Cost: 250250
Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program
1
The attributes for this house are:
Age: 1
Type: Attached
Rooms: 4
Cost: 150500
```

Question 3, testhouse.cpp

terhouse.cpp

house.cpp

house.h

Miscellaneous Files - No Configurations

(Global Scope)

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

```
}
cout << "Rooms Number: ";
cin >> rooms; // 'rooms' input
cout << "Cost: ";
cin >> cost; // 'cost' input

// first loop means i is 1, thus storing the data in h1
if (i == 1) {
    h1.ageH(age); // set age of the 'house'
    h1.typeH(type); // set the type of the 'house'
    h1.roomsH(rooms); // set the rooms of the 'house'
    h1.costH(cost); // set the cost of the 'house'
}
// second loop means i is 2, thus storing the data in h2
else {
    h2.ageH(age); // set the age of the 'house'
    h2.typeH(type); // set the type of the 'house'
    h2.roomsH(rooms); // set the rooms of the 'house'
    h2.costH(cost); // set the cost of the 'house'
}
}

// loop as long as user doesn't input '5'
// this loop allows for user to interact with the member functions
while (checkNum != 6) {
    cout << "\nEnter a number to perform one of the following commands:";
    cout << "\n1. Display the Information of House One";
    cout << "\n2. Display the Information of House Two";
    cout << "\n3. Edit an attribute of House One";
    cout << "\n4. Edit an attribute of House Two";
    cout << "\n5. Display the price estimate of a house in the future";
    cout << "\n6. Exit Program\n\n";

    // switch-case performing the selected function
    cin >> checkNum;
    switch (checkNum) {
        case 1: // prints/get function for all attributes
            h1.display();
```

Select Microsoft Visual Studio Debug Console

Age: 1

Type: Attached

Rooms: 4

Cost: 150500

Enter a number to perform one of the following commands:

1. Display the Information of House One

2. Display the Information of House Two

3. Edit an attribute of House One

4. Edit an attribute of House Two

5. Display the price estimate of a house in the future

6. Exit Program

2

The attributes for this house are:

Age: 20

Type: Detached

Rooms: 8

Cost: 250250

Enter a number to perform one of the following commands:

1. Display the Information of House One

2. Display the Information of House Two

3. Edit an attribute of House One

4. Edit an attribute of House Two

5. Display the price estimate of a house in the future

6. Exit Program

4

Enter a number corresponding an attribute to edit

1. Age

2. Type

3. Rooms

4. Cost

1

Age: 1000

Enter a number to perform one of the following commands:

1. Display the Information of House One

2. Display the Information of House Two

3. Edit an attribute of House One

4. Edit an attribute of House Two

5. Display the price estimate of a house in the future

6. Exit Program

4

Enter a number corresponding an attribute to edit

1. Age

2. Type

3. Rooms

4. Cost

2

Type: ('Detached', 'Semi-Attached', 'Attached') hgfd

Try again...

Type: ertyhjbfg

Try again...

121%

No issues found

Question 3, testhouse.cpp

```
testhouse.cpp x house.cpp house.h (Global Scope)
Miscellaneous Files - No Configurations

73 case 1: // prints/get function for all attributes
74 h1.display();
75 break;
76 case 2: // print/get function for all attributes
77 h2.display();
78 break;
79 case 3: // modify/set function for all attributes of House 1
80 cout << "Enter a number corresponding an attribute to edit";
81 cout << "\n1. Age";
82 cout << "\n2. Type";
83 cout << "\n3. Rooms";
84 cout << "\n4. Cost\n";
85
86 // switch-case for editing a specific attribute
87 cin >> editNum;
88 switch (editNum) {
89 case 1:
90     cout << "Age: ";
91     cin >> age;
92     h1.ageH(age); // set category of the device
93     break;
94 case 2:
95     cin >> type;
96     cout << "Type: ('Detached', 'Semi-Attached', 'Attached') ";
97     cin >> type; // 'type' input
98     // repeat until the type corresponds to one of the option
99     while (type != "Detached" && type != "Semi-Attached" && type
100         cout << "Try again...\nType: ";
101         cin >> type;
102     }
103     h1.typeH(type); // set the color of the device
104     break;
105 case 3:
106     cout << "Rooms: ";
107     cin >> rooms;
108     h1.roomsH(rooms); // set the model number of the device
109     break;
110 case 4:
111     cout << "Cost: ";
112     cin >> cost;
113     h1.costH(cost); // set the price estimate of the house
114     break;
115 }
116 break;
117 }
118 }
119 }
120 }
```

Select Microsoft Visual Studio Debug Console

```
2
Type: ('Detached', 'Semi-Attached', 'Attached') hgfd
Try again...
Type: ertyhbjbgg
Try again...
Type: Semi-Attached

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

2
The attributes for this house are:
Age: 1000
Type: Semi-Attached
Rooms: 8
Cost: 250250

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

5
Type: ('Detached', 'Semi-Attached', 'Attached') hgfdscfvgh
Try again...
Type: Attached
Enter the years from now: 25
The cost estimate of this house is 227120

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

6
You have exited the program...
E:\vsCode\Project Location\COEN 243 Assignments\Assignment
d with code 0.
To automatically close the console when debugging stops, en
le when debugging stops.
Press any key to close this window . . .
```

Question 3, testhouse.cpp

```
testhouse.cpp  house.cpp  house.h
Miscellaneous Files - No Configurations  (Global Scope)

109         break;
110     case 4:
111         cout << "Cost: ";
112         cin >> cost;
113         h1.costH(cost); // set the status of the device
114         break;
115     }
116     break;
117     case 4: // modify/set function for all attributes of House 2
118         cout << "Enter a number corresponding an attribute to edit";
119         cout << "\n1. Age";
120         cout << "\n2. Type";
121         cout << "\n3. Rooms";
122         cout << "\n4. Cost\n";
123
124         // switch-case for editing a specific attribute
125         cin >> editNum;
126         switch (editNum) {
127         case 1:
128             cout << "Age: ";
129             cin >> age;
130             h2.ageH(age); // set category of the device
131             break;
132         case 2:
133             cout << "Type: ('Detached', 'Semi-Attached', 'Attached') ";
134             cin >> type; // 'type' input
135             // repeat until the type corresponds to one of the option
136             while (type != "Detached" && type != "Semi-Attached" && type != "Attached") {
137                 cout << "Try again...\nType: ";
138                 cin >> type;
139             }
140             h2.typeH(type); // set the color of the device
141             break;
142         case 3:
143             cout << "Rooms: ";
144             cin >> rooms;
145             h2.roomsH(rooms); // set the model number of the device
146             break;
```


Question 3, testhouse.cpp

```
testhouse.cpp x house.cpp house.h
Miscellaneous Files - No Configurations (Global Scope)

142 case 3:
143     cout << "Rooms: ";
144     cin >> rooms;
145     h2.roomsH(rooms); // set the model number of the device
146     break;
147 case 4:
148     cout << "Cost: ";
149     cin >> cost;
150     h2.costH(cost); // set the status of the device
151     break;
152 }
153 break;
154 case 5: // 'estimatePrice()' function call
155     cout << "Type: ('Detached', 'Semi-Attached', 'Attached') ";
156     cin >> type; // 'type' input
157     // repeat until the type corresponds to one of the option
158     while (type != "Detached" && type != "Semi-Attached" && type != "Attached") {
159         cout << "Try again...\nType: ";
160         cin >> type;
161     }
162     cout << "Enter the years from now: ";
163     cin >> yearEstPri; // number of years looking into the future
164     cout << "The cost estimate of this house is ";
165     cout << h.estimatePrice(type, yearEstPri) << endl; // function call
166     break;
167 case 6: // exits program
168     cout << "You have exited the program...";
169     break;
170 }
171 }
172 return 0;
173 }
```

Question 3, house.cpp

```
testhouse.cpp | house.cpp | house.h
Miscellaneous Files - No Configurations | -> House

1 // Assignment 4, Question 3, Week 10
2 // Hei Wang Andre Law, 4017 5600
3 // Program: user-defined class "House" defined by attributes
4 // (cpp file for implementing the member functions of the class)
5
6 #include<iostream>
7 #include <string>
8 #include "house.h"
9 using namespace std;
10
11 // class implementation for 'House'
12 House::House() {
13     age = 0;
14     type = "";
15     rooms = 0;
16     cost = 0.0;
17 }
18
19 // function estimating the price of a future house
20 double House::estimatePrice(string typeH, int year) {
21     double appRateI; // appreciation rate within 5 years (Initial)
22     double appRateF; // appreciation rate after 5 years (Final)
23     int currentPri; // current house price
24     double estPri; // final value of calculated price estimate
25
26     // checks which type of house it is
27     if (typeH == "Detached") {
28         appRateI = 0.02;
29         appRateF = 0.02;
30         currentPri = 200000;
31     }
32     else if (typeH == "Semi-Attached") {
33         appRateI = 0.02;
34         appRateF = 0.03;
35         currentPri = 150000;
36     }
37     else {
38         appRateI = 0.01;
39     }
40 }
41
42 // function to display house information
43 void House::displayInfo() const {
44     cout << "House Information:" << endl;
45     cout << "Age: " << age << endl;
46     cout << "Type: (" << type << ", 'Detached', 'Semi-Attached', 'Attached') " << type << endl;
47     cout << "Rooms Number: " << rooms << endl;
48     cout << "Cost: " << cost << endl;
49 }
50
51 // function to edit house attribute
52 void House::editAttribute(string attr, int value) {
53     if (attr == "age") {
54         age = value;
55     }
56     else if (attr == "rooms") {
57         rooms = value;
58     }
59     else if (attr == "type") {
60         type = value;
61     }
62 }
63
64 // function to estimate price of a house in the future
65 void House::estimateFuturePrice(int years, double& estPri) const {
66     double currentPri;
67     double appRateI;
68     double appRateF;
69     double estPri;
70
71     // get current house price
72     cout << "Enter the years from now: ";
73     int years;
74     while (years < 0) {
75         cout << "Invalid input. Please enter a non-negative integer." << endl;
76         cout << "Enter the years from now: ";
77     }
78     currentPri = cost;
79     appRateI = 0.02;
80     appRateF = 0.02;
81
82     // calculate the price estimate
83     for (int i = 0; i < years; i++) {
84         currentPri = currentPri * (1 + appRateI);
85     }
86     appRateI = appRateF;
87     estPri = currentPri;
88 }
89
90 // function to display price estimate of a house in the future
91 void House::displayEstimatePrice(int years, double estPri) const {
92     cout << "The cost estimate of this house is " << estPri << endl;
93 }
94
95 // function to exit the program
96 void House::exitProgram() {
97     cout << "Exit Program" << endl;
98 }
```

Select Microsoft Visual Studio Debug Console

Enter the information for 'house' number 1

Age: 1
Type: ('Detached', 'Semi-Attached', 'Attached') Attached
Rooms Number: 12
Cost: 123456

Enter the information for 'house' number 2

Age: 4
Type: ('Detached', 'Semi-Attached', 'Attached') Detached
Rooms Number: 4
Cost: 654321

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

5
Type: ('Detached', 'Semi-Attached', 'Attached') Attached
Enter the years from now: 10
The cost estimate of this house is 218202

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

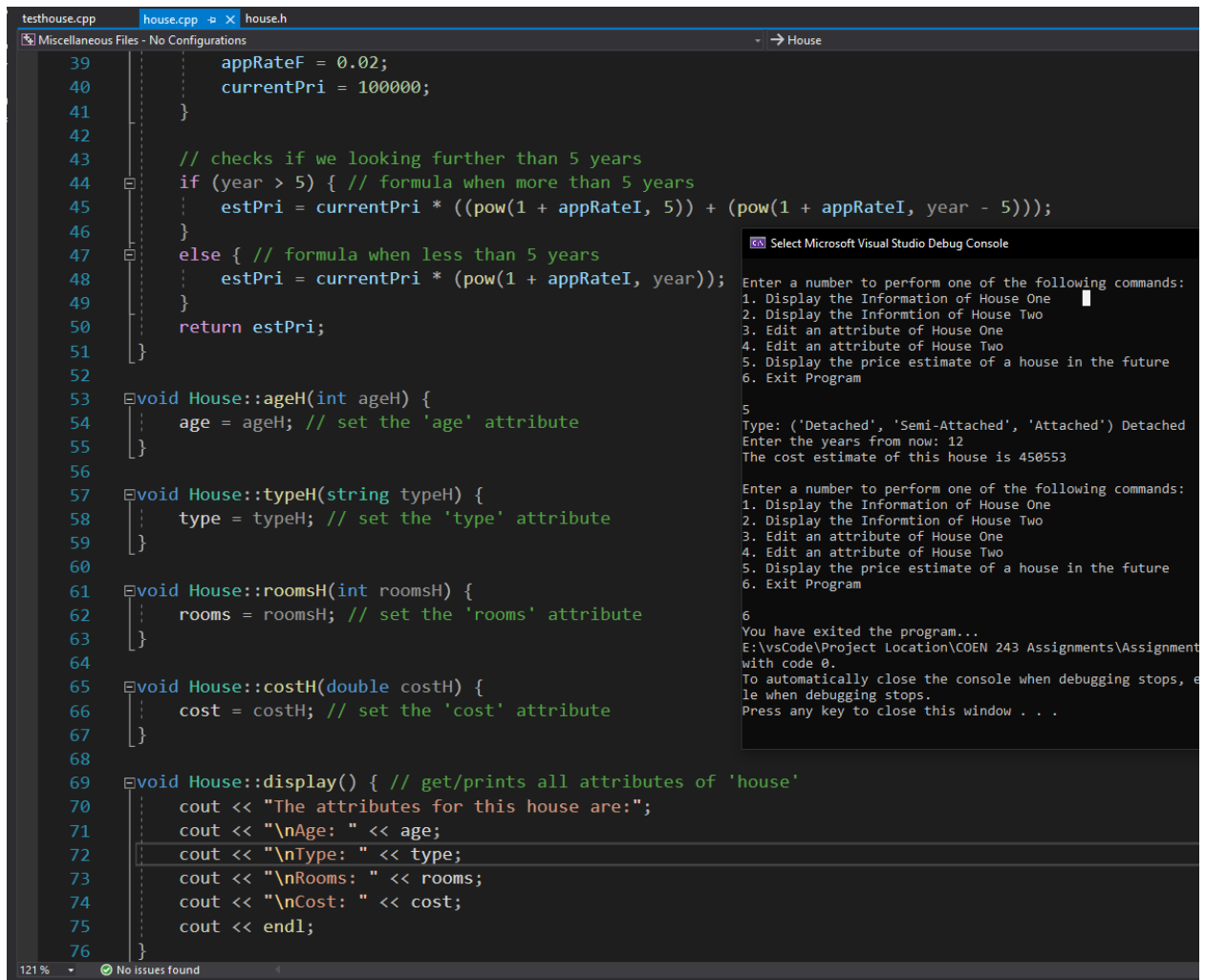
5
Type: ('Detached', 'Semi-Attached', 'Attached') Attached
Enter the years from now: 4
The cost estimate of this house is 104060

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

5
Type: ('Detached', 'Semi-Attached', 'Attached') Detached
Enter the years from now: 2
The cost estimate of this house is 208080

Enter a number to perform one of the following commands:
1. Display the Information of House One

Question 3, house.cpp



```
testhouse.cpp  house.cpp  house.h
Miscellaneous Files - No Configurations
39      appRateF = 0.02;
40      currentPri = 100000;
41  }
42
43      // checks if we looking further than 5 years
44      if (year > 5) { // formula when more than 5 years
45          estPri = currentPri * ((pow(1 + appRateI, 5)) + (pow(1 + appRateI, year - 5)));
46      }
47      else { // formula when less than 5 years
48          estPri = currentPri * (pow(1 + appRateI, year));
49      }
50      return estPri;
51  }
52
53  void House::ageH(int ageH) {
54      age = ageH; // set the 'age' attribute
55  }
56
57  void House::typeH(string typeH) {
58      type = typeH; // set the 'type' attribute
59  }
60
61  void House::roomsH(int roomsH) {
62      rooms = roomsH; // set the 'rooms' attribute
63  }
64
65  void House::costH(double costH) {
66      cost = costH; // set the 'cost' attribute
67  }
68
69  void House::display() { // get/prints all attributes of 'house'
70      cout << "The attributes for this house are:";
71      cout << "\nAge: " << age;
72      cout << "\nType: " << type;
73      cout << "\nRooms: " << rooms;
74      cout << "\nCost: " << cost;
75      cout << endl;
76  }
```

Select Microsoft Visual Studio Debug Console

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

5
Type: ('Detached', 'Semi-Attached', 'Attached') Detached
Enter the years from now: 12
The cost estimate of this house is 450553

Enter a number to perform one of the following commands:
1. Display the Information of House One
2. Display the Information of House Two
3. Edit an attribute of House One
4. Edit an attribute of House Two
5. Display the price estimate of a house in the future
6. Exit Program

6
You have exited the program...
E:\vsCode\Project Location\COEN 243 Assignments\Assignment
with code 0.
To automatically close the console when debugging stops, e
le when debugging stops.
Press any key to close this window . . .

121 % No issues found

Question 3, house.h

```
testhouse.cpp  house.cpp  house.h  X
Miscellaneous Files - No Configurations  House

1  // Assignment 4, Question 3, Week 10
2  // Hei Wang Andre Law, 4017 5600
3  // Program: user-defined class "House" defined by attributes
4  // (header file containing the specification of the class)
5
6  #include <string>
7  using namespace std;
8
9  // class definition for Edevice
10 class House {
11 private: // data members
12     int age;
13     string type;
14     int rooms;
15     double cost;
16
17 public: // member function
18     House(); // constructor
19     double estimatePrice(string typeH, int year); // estimation function call
20     void ageH(int age); // attributes
21     void typeH(string type); // attributes
22     void roomsH(int rooms); // attributes
23     void costH(double cost); // attributes
24     void display(); // prints/get/display the current house attributes
25 }; // end of class Edevice
```