



Topic : Online Bill and Event Reminder System

Group no : MLB\_WE\_01.01\_G1\_08

Campus : **Malabe** / Metro / Matara / Kandy / Kurunegala / Kandy / Jaffna

Submission Date : 19 /05/2022

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21108822	Abeykoon M W T H B	071 155 5722
IT21044304	Kariyapperuma K A D C A	075 563 7447
IT21224966	Dilrukshi A G T	071 361 6158
IT21237522	Thathsarani H A N N	077 752 3879
IT21227486	Ruhunage S D P R	071 110 8805

# Online Bill and Event Reminder System

Customer can visit the system. In system view various payment services. These are insurance, utility and mobile. If customer prefer to pay bills, then the customer has to register to the system. At the registration, customer enters their details and accept the terms and conditions. Registered customer is a customer. Then registered customer can add to favourites. These are mobile bills, electricity bills and water bills. Also registered customer can add reminders and remove reminders. If registered customer wants to changed account details, then they can update the profile. Furthermore, registered customer can make bill payments by visa, master, or PayPal. Customer care person contact customers, create reports, and provide information and solutions. When customer care person contacts with customers, these customers give feedbacks via messages or via emails or via calls. Web developer is a kind of customer care person. Web developer can monitor website traffic, troubleshoot web site problems, changed web site structure, maintain website and update website. Administrator is a kind of web developer. Administrator can manage payment services, add new events, remove events and update database. If need to change user access, then the administrator can manage users by add users and remove users.

## Classes:

1. Customer
2. Registered Customer
3. Customer Care Person
4. Web Developer
5. Administrator
6. Bill
7. Event
8. Payment

# CRC Cards

1.

Customer Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"><li>Visit the system</li></ul>	

2.

Registered Customer Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"><li>Register Details</li><li>Pay bills</li><li>Visit the system</li><li>Add favourites</li><li>Add reminders</li><li>Remove reminders</li><li>Update profile</li><li>Give feedbacks</li></ul>	<p>Bill class</p> <p>Bill class</p> <p>Bill class, Event class</p> <p>Bill class, Event class</p> <p>Customer care person class</p>

3.

Customer Care Person Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"><li>Contact customers</li><li>Create reports</li><li>Provide information</li><li>Provide solution</li></ul>	<p>Customer class, Registered Customer class</p> <p>Bill class, Event class</p>

4.

Web developer Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"> <li>• Contact customers</li> <li>• Create reports</li> <li>• Provide information</li> <li>• Provide solutions</li> <li>• Monitor website traffic</li> <li>• Troubleshoot website problems</li> <li>• Change website structure</li> <li>• Maintain website</li> <li>• Update website</li> </ul>	Customer class, Registered Customer class Bill class, Event class

5.

Administrator Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"> <li>• Contact customers</li> <li>• Create reports</li> <li>• Provide information</li> <li>• Provide solutions</li> <li>• Monitor website traffic</li> <li>• Troubleshoot website problems</li> <li>• Change website structure</li> <li>• Maintain website</li> <li>• Update website</li> <li>• Manage payment services</li> <li>• Add and remove events</li> <li>• Update database</li> <li>• Manage user access</li> </ul>	Customer class, Registered Customer class Bill class, Event class  Payment class Event class

6.

Bill Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"><li>• Details of bills</li></ul>	

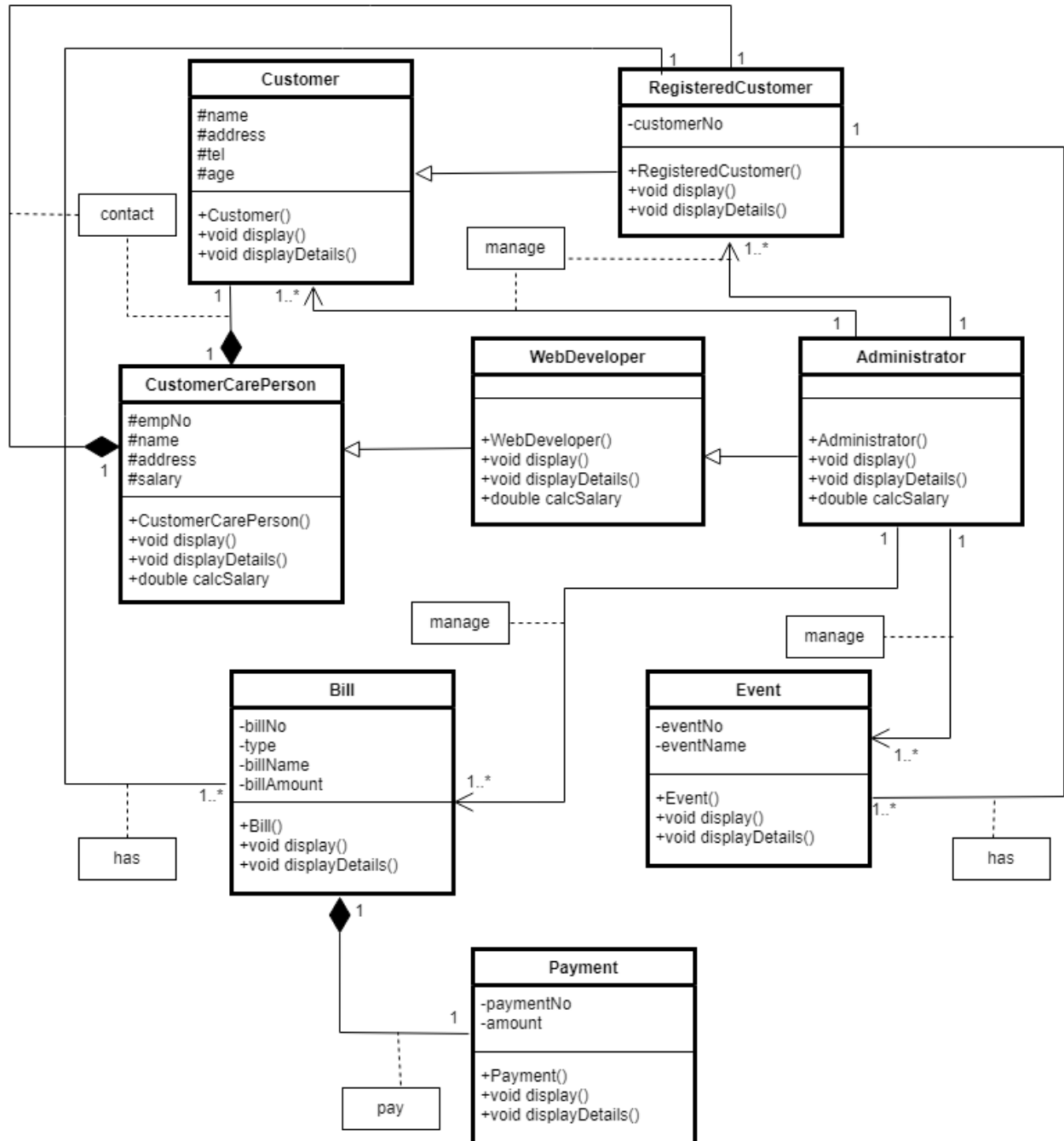
7.

Event Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"><li>• Details of events</li></ul>	

8.

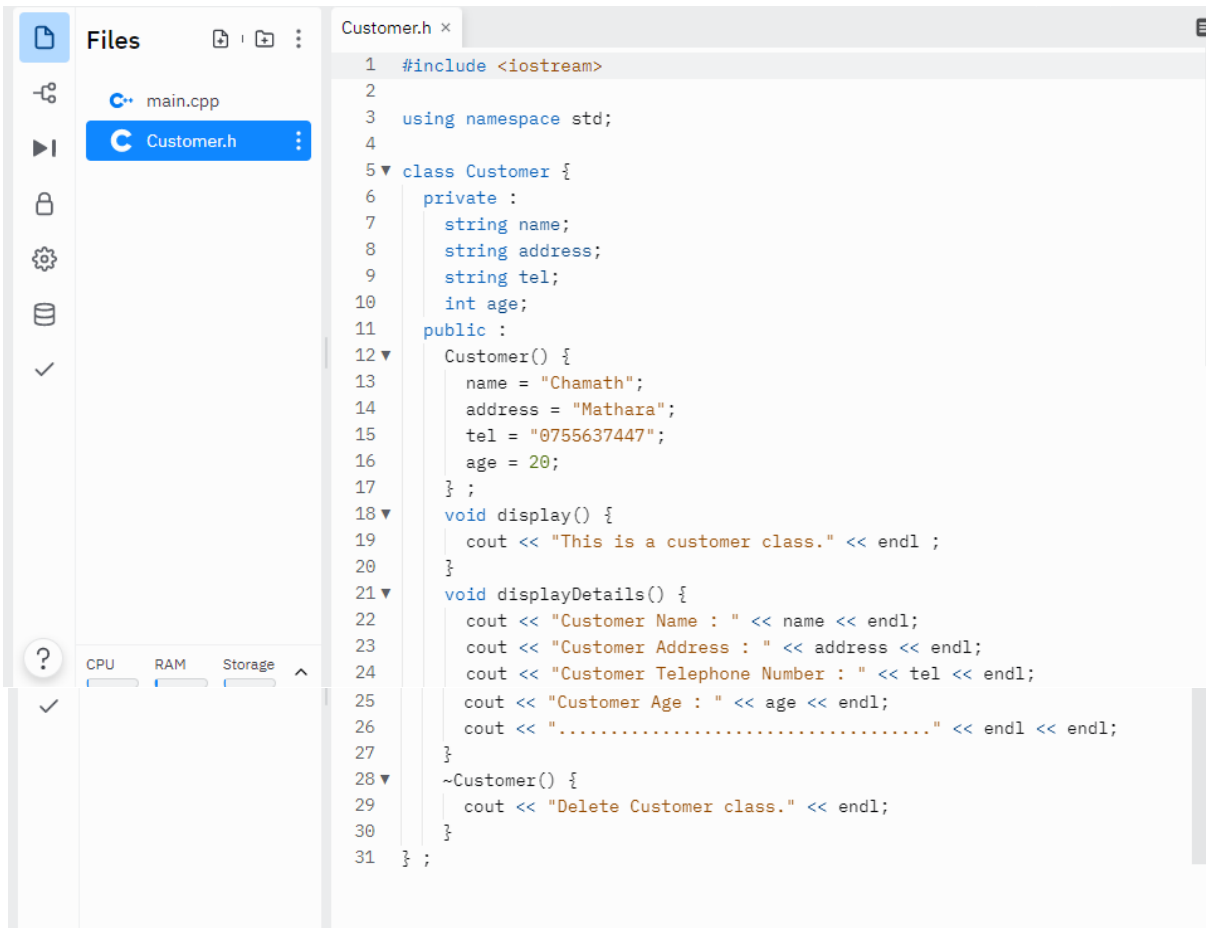
Payment Class	
<u>Responsibilities</u>	<u>Collaborations</u>
<ul style="list-style-type: none"><li>• Details of payments</li><li>• Validate of payments</li></ul>	

# Exercise: 01



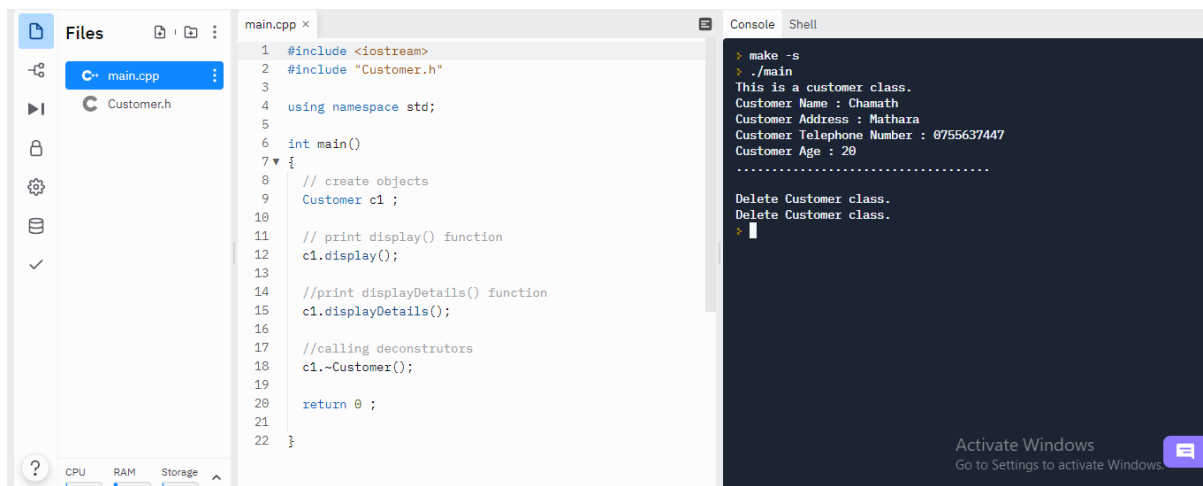
# Exercise: 02

## 01. Customer.h file (created by Kariyapperuma K A D C A)



```
1 #include <iostream>
2
3 using namespace std;
4
5 class Customer {
6     private :
7         string name;
8         string address;
9         string tel;
10        int age;
11    public :
12        Customer() {
13            name = "Chamath";
14            address = "Mathara";
15            tel = "0755637447";
16            age = 20;
17        } ;
18        void display() {
19            cout << "This is a customer class." << endl ;
20        }
21        void displayDetails() {
22            cout << "Customer Name : " << name << endl;
23            cout << "Customer Address : " << address << endl;
24            cout << "Customer Telephone Number : " << tel << endl;
25            cout << "Customer Age : " << age << endl;
26            cout << "....." << endl << endl;
27        }
28        ~Customer() {
29            cout << "Delete Customer class." << endl;
30        }
31    } ;
```

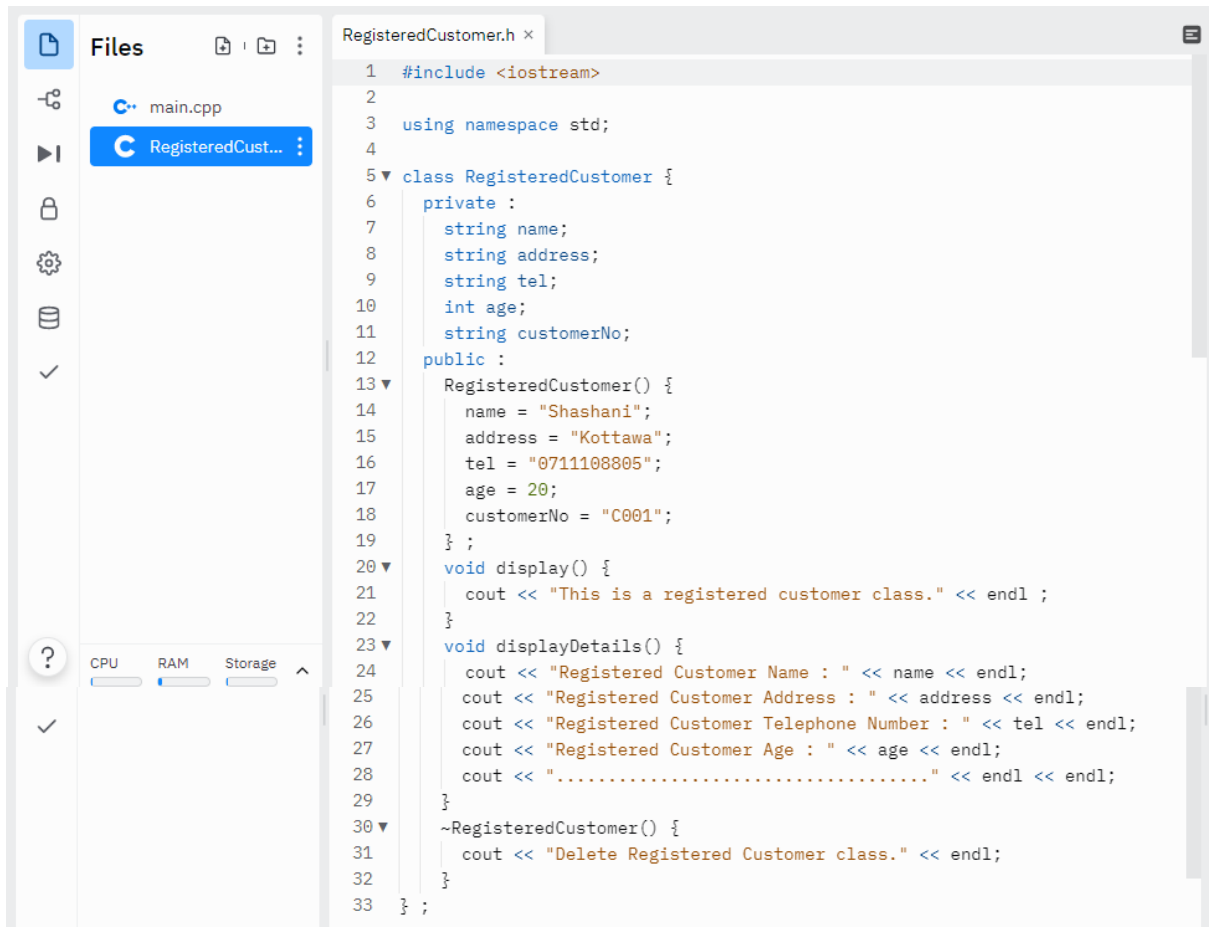
## Customer.cpp file (created by Kariyapperuma K A D C A)



```
1 #include <iostream>
2 #include "Customer.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     Customer c1 ;
10
11     // print display() function
12     c1.display();
13
14     //print displayDetails() function
15     c1.displayDetails();
16
17     //calling deconstructors
18     c1.~Customer();
19
20     return 0 ;
21
22 }
```

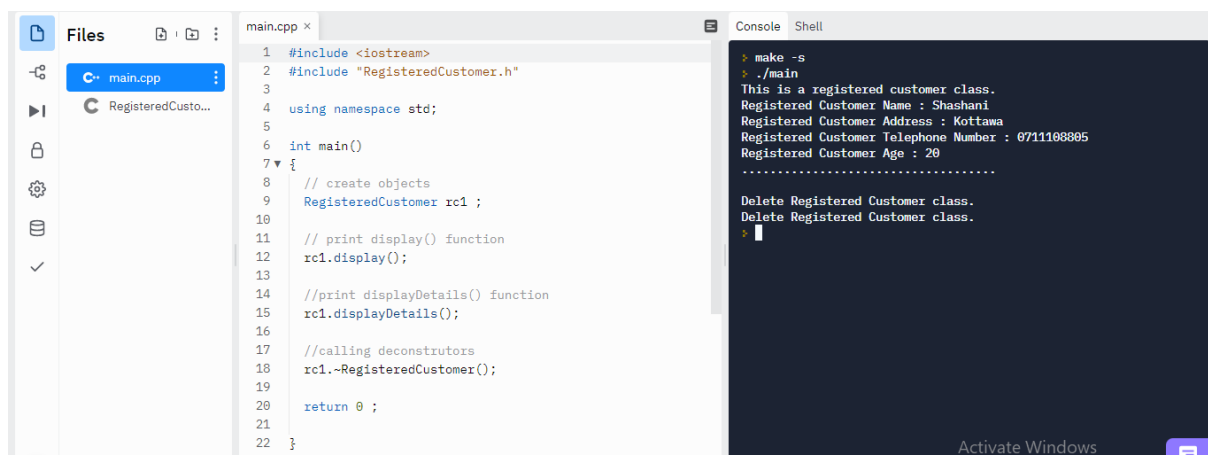
```
> make -s
> ./main
This is a customer class.
Customer Name : Chamath
Customer Address : Mathara
Customer Telephone Number : 0755637447
Customer Age : 20
.....
Delete Customer class.
Delete Customer class.
```

## 02. RegisteredCustomer.h file (created by Ruhunage S D P R)



```
1 #include <iostream>
2
3 using namespace std;
4
5 class RegisteredCustomer {
6     private :
7         string name;
8         string address;
9         string tel;
10        int age;
11        string customerNo;
12    public :
13        RegisteredCustomer() {
14            name = "Shashani";
15            address = "Kottawa";
16            tel = "0711108805";
17            age = 20;
18            customerNo = "C001";
19        } ;
20        void display() {
21            cout << "This is a registered customer class." << endl ;
22        }
23        void displayDetails() {
24            cout << "Registered Customer Name : " << name << endl;
25            cout << "Registered Customer Address : " << address << endl;
26            cout << "Registered Customer Telephone Number : " << tel << endl;
27            cout << "Registered Customer Age : " << age << endl;
28            cout << "....." << endl << endl;
29        }
30        ~RegisteredCustomer() {
31            cout << "Delete Registered Customer class." << endl;
32        }
33    } ;
```

## RegisteredCustomer.cpp file (created by Ruhunage S D P R)



```
1 #include <iostream>
2 #include "RegisteredCustomer.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     RegisteredCustomer rc1 ;
10
11     // print display() function
12     rc1.display();
13
14     //print displayDetails() function
15     rc1.displayDetails();
16
17     //calling destructors
18     rc1.~RegisteredCustomer();
19
20     return 0 ;
21 }
22 }
```

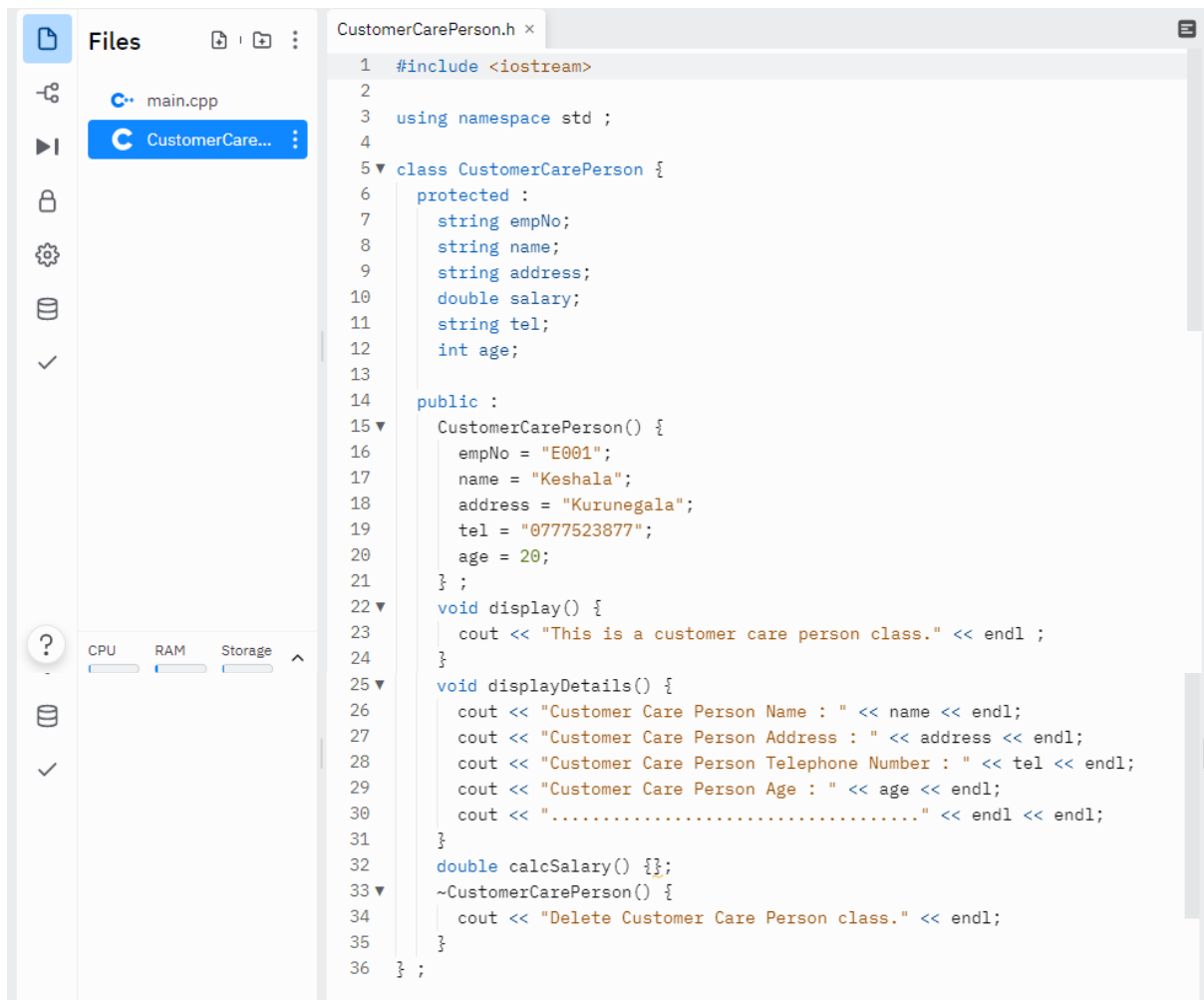
Console Output:

```
> make -s
> ./main
This is a registered customer class.
Registered Customer Name : Shashani
Registered Customer Address : Kottawa
Registered Customer Telephone Number : 0711108805
Registered Customer Age : 20
.....

Delete Registered Customer class.
Delete Registered Customer class.
```

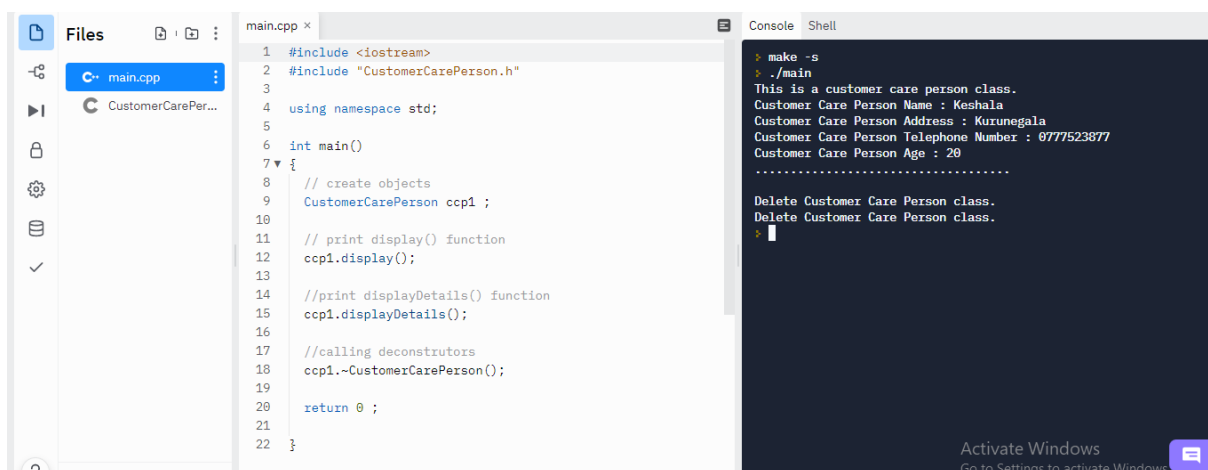


### 03. CustomerCarePerson.h file (created by Thathsarani H A N N)



```
1 #include <iostream>
2
3 using namespace std ;
4
5 class CustomerCarePerson {
6     protected :
7         string empNo;
8         string name;
9         string address;
10        double salary;
11        string tel;
12        int age;
13
14    public :
15        CustomerCarePerson() {
16            empNo = "E001";
17            name = "Keshala";
18            address = "Kurunegala";
19            tel = "0777523877";
20            age = 20;
21        } ;
22        void display() {
23            cout << "This is a customer care person class." << endl ;
24        }
25        void displayDetails() {
26            cout << "Customer Care Person Name : " << name << endl;
27            cout << "Customer Care Person Address : " << address << endl;
28            cout << "Customer Care Person Telephone Number : " << tel << endl;
29            cout << "Customer Care Person Age : " << age << endl;
30            cout << "....." << endl << endl;
31        }
32        double calcSalary() {} ;
33        ~CustomerCarePerson() {
34            cout << "Delete Customer Care Person class." << endl;
35        }
36    } ;
```

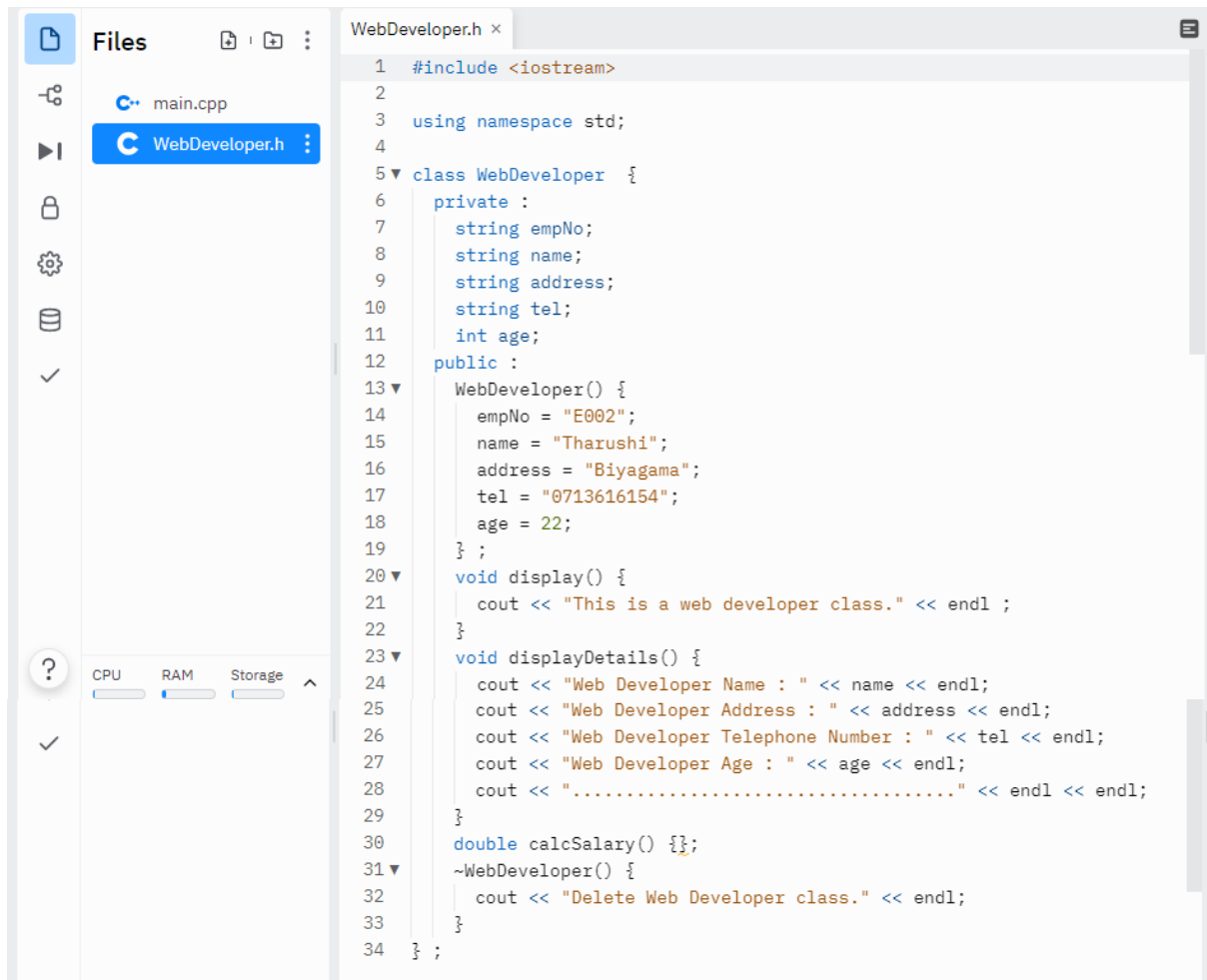
### CustomerCarePerson.cpp file (created by Thathsarani H A N N)



```
1 #include <iostream>
2 #include "CustomerCarePerson.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     CustomerCarePerson ccp1 ;
10
11     // print display() function
12     ccp1.display();
13
14     //print displayDetails() function
15     ccp1.displayDetails();
16
17     //calling deonstrutors
18     ccp1.~CustomerCarePerson();
19
20     return 0 ;
21 }
22 }
```

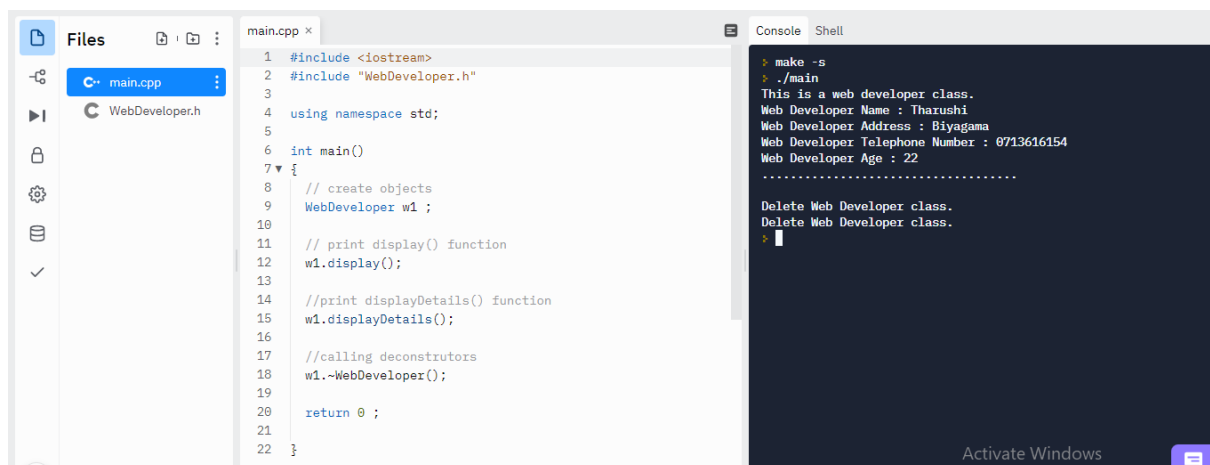
```
> make -s
> ./main
This is a customer care person class.
Customer Care Person Name : Keshala
Customer Care Person Address : Kurunegala
Customer Care Person Telephone Number : 0777523877
Customer Care Person Age : 20
.....
Delete Customer Care Person class.
Delete Customer Care Person class.
>
```

## 04. WebDeveloper.h file (created by Dilrukshi A G T)



```
1 #include <iostream>
2
3 using namespace std;
4
5 class WebDeveloper {
6     private :
7         string empNo;
8         string name;
9         string address;
10        string tel;
11        int age;
12    public :
13        WebDeveloper() {
14            empNo = "E002";
15            name = "Tharushi";
16            address = "Biyagama";
17            tel = "0713616154";
18            age = 22;
19        } ;
20        void display() {
21            cout << "This is a web developer class." << endl ;
22        }
23        void displayDetails() {
24            cout << "Web Developer Name : " << name << endl;
25            cout << "Web Developer Address : " << address << endl;
26            cout << "Web Developer Telephone Number : " << tel << endl;
27            cout << "Web Developer Age : " << age << endl;
28            cout << "....." << endl << endl;
29        }
30        double calcSalary() {} ;
31        ~WebDeveloper() {
32            cout << "Delete Web Developer class." << endl;
33        }
34    } ;
```

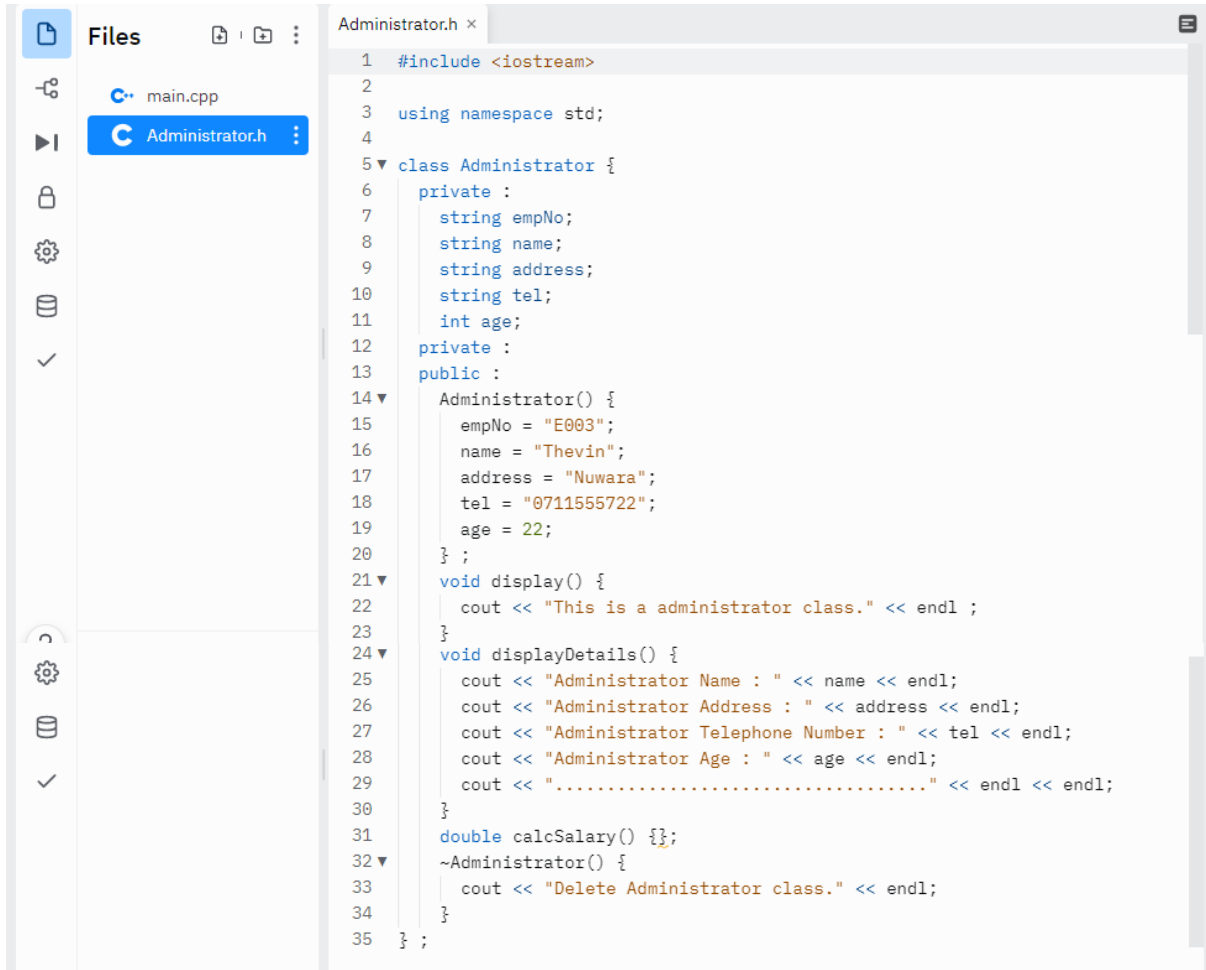
## WebDeveloper.cpp file (created by Dilrukshi A G T)



```
1 #include <iostream>
2 #include "WebDeveloper.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     WebDeveloper w1 ;
10
11     // print display() function
12     w1.display();
13
14     //print displayDetails() function
15     w1.displayDetails();
16
17     //calling destructors
18     w1.~WebDeveloper();
19
20     return 0 ;
21 }
22 }
```

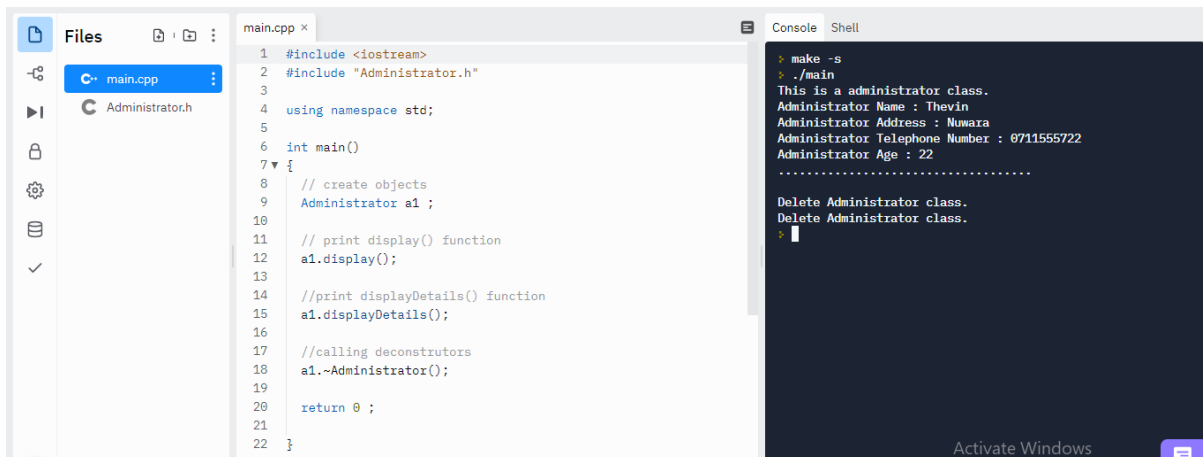
```
> make -s
> ./main
This is a web developer class.
Web Developer Name : Tharushi
Web Developer Address : Biyagama
Web Developer Telephone Number : 0713616154
Web Developer Age : 22
.....
Delete Web Developer class.
Delete Web Developer class.
>
```

## 05. Administrator.h file (created by Abeykoon M W T H B)



```
1 #include <iostream>
2
3 using namespace std;
4
5 class Administrator {
6     private :
7         string empNo;
8         string name;
9         string address;
10        string tel;
11        int age;
12    private :
13    public :
14        Administrator() {
15            empNo = "E003";
16            name = "Thevin";
17            address = "Nuwara";
18            tel = "0711555722";
19            age = 22;
20        } ;
21        void display() {
22            cout << "This is a administrator class." << endl ;
23        }
24        void displayDetails() {
25            cout << "Administrator Name : " << name << endl;
26            cout << "Administrator Address : " << address << endl;
27            cout << "Administrator Telephone Number : " << tel << endl;
28            cout << "Administrator Age : " << age << endl;
29            cout << "....." << endl << endl;
30        }
31        double calcSalary() { };
32        ~Administrator() {
33            cout << "Delete Administrator class." << endl;
34        }
35    } ;
```

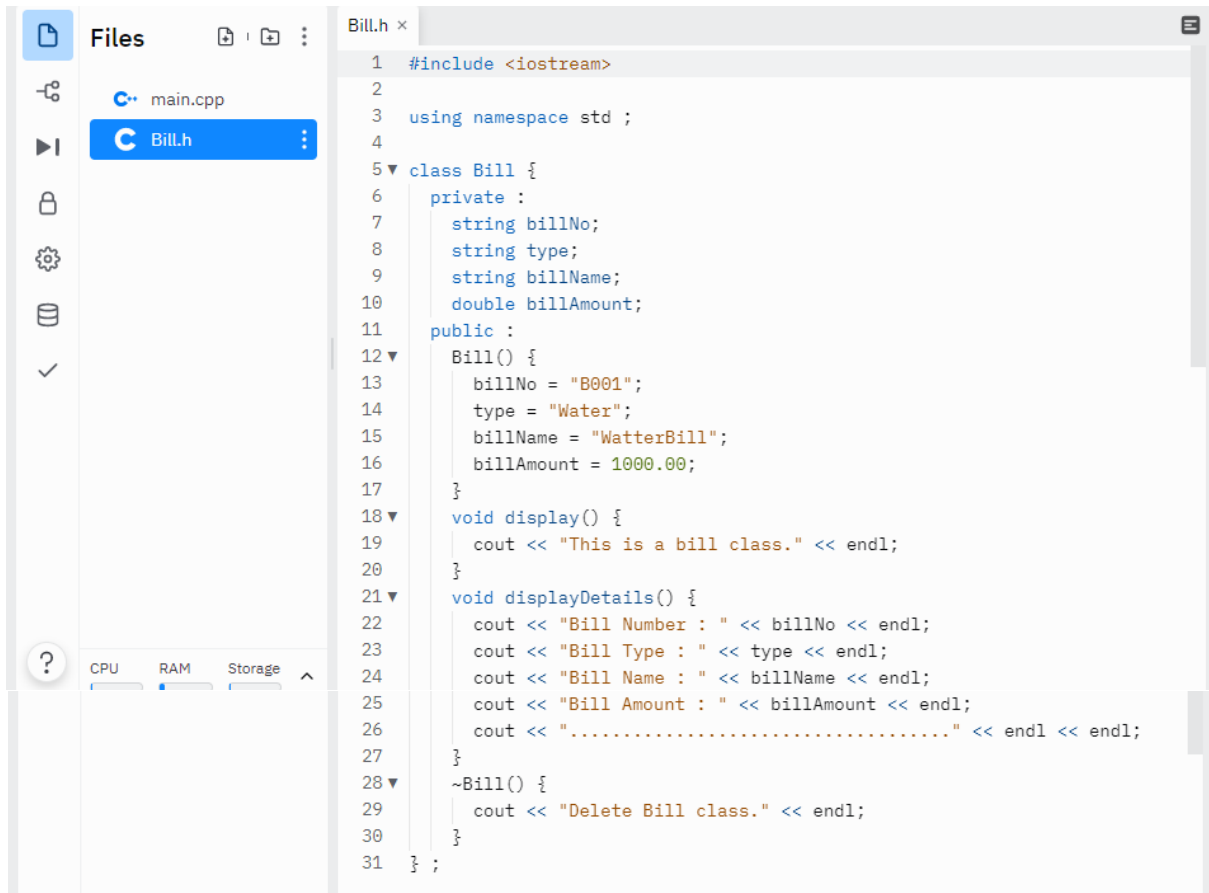
## Administrator.cpp file (created by Abeykoon M W T H B)



```
1 #include <iostream>
2 #include "Administrator.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     Administrator a1 ;
10
11     // print display() function
12     a1.display();
13
14     //print displayDetails() function
15     a1.displayDetails();
16
17     //calling deconstructors
18     a1.~Administrator();
19
20     return 0 ;
21
22 }
```

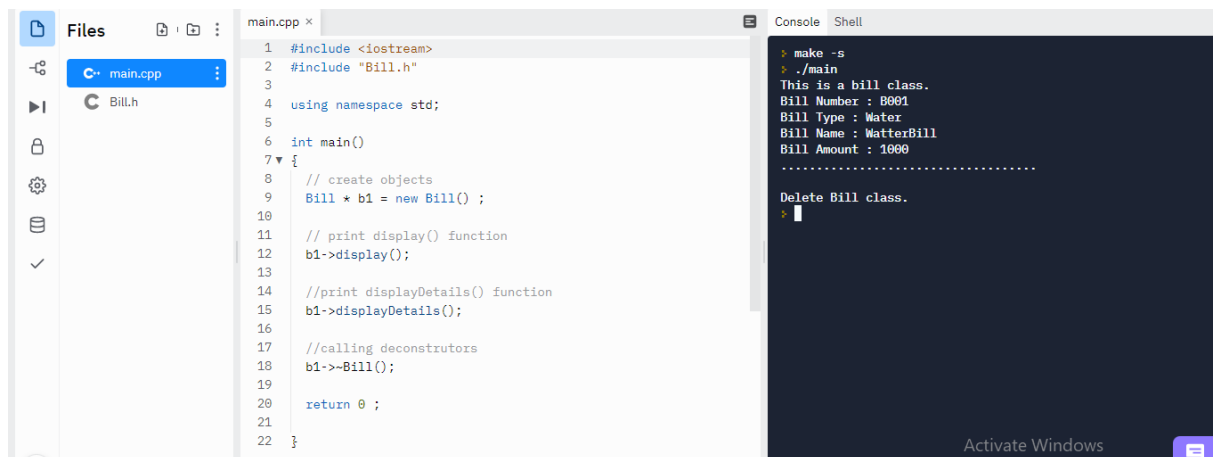
```
> make -s
> ./main
This is a administrator class.
Administrator Name : Thevin
Administrator Address : Nuwara
Administrator Telephone Number : 0711555722
Administrator Age : 22
.....
Delete Administrator class.
Delete Administrator class.
>
```

## 06. Bill.h file (created by Ruhunage S D P R)



```
1 #include <iostream>
2
3 using namespace std ;
4
5 class Bill {
6     private :
7         string billNo;
8         string type;
9         string billName;
10        double billAmount;
11    public :
12        Bill() {
13            billNo = "B001";
14            type = "Water";
15            billName = "WatterBill";
16            billAmount = 1000.00;
17        }
18        void display() {
19            cout << "This is a bill class." << endl;
20        }
21        void displayDetails() {
22            cout << "Bill Number : " << billNo << endl;
23            cout << "Bill Type : " << type << endl;
24            cout << "Bill Name : " << billName << endl;
25            cout << "Bill Amount : " << billAmount << endl;
26            cout << "....." << endl << endl;
27        }
28        ~Bill() {
29            cout << "Delete Bill class." << endl;
30        }
31    } ;
```

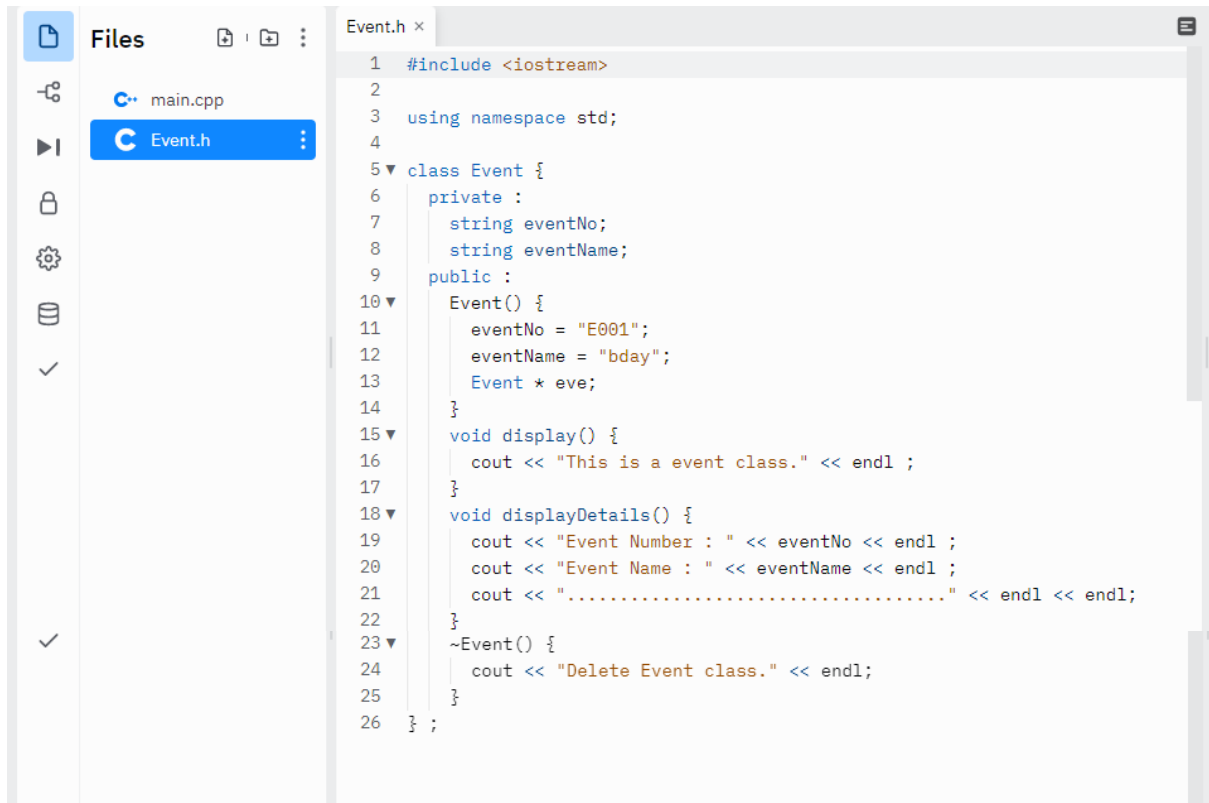
## Bill.cpp file (created by Ruhunage S D P R)



```
1 #include <iostream>
2 #include "Bill.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     Bill * b1 = new Bill() ;
10
11     // print display() function
12     b1->display();
13
14     //print displayDetails() function
15     b1->displayDetails();
16
17     //calling destructors
18     b1->~Bill();
19
20     return 0 ;
21 }
22 }
```

```
> make -s
> ./main
This is a bill class.
Bill Number : B001
Bill Type : Water
Bill Name : WatterBill
Bill Amount : 1000
.....
Delete Bill class.
>
```

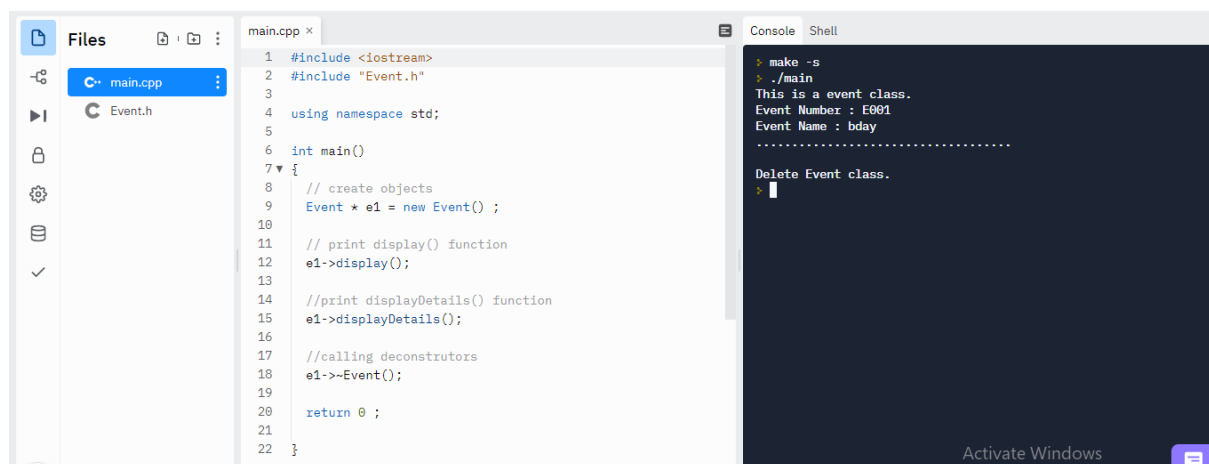
## 07. Event.h file (created by Dilrukshi A G T)



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows two files: main.cpp and Event.h. The Event.h file is selected and its content is displayed in the code editor. The code defines a class Event with private attributes eventNo and eventName, and a public attribute eve. It includes a constructor, a destructor, and two display methods.

```
1 #include <iostream>
2
3 using namespace std;
4
5 class Event {
6     private :
7         string eventNo;
8         string eventName;
9     public :
10        Event() {
11            eventNo = "E001";
12            eventName = "bdays";
13            Event * eve;
14        }
15        void display() {
16            cout << "This is a event class." << endl ;
17        }
18        void displayDetails() {
19            cout << "Event Number : " << eventNo << endl ;
20            cout << "Event Name : " << eventName << endl ;
21            cout << "....." << endl << endl;
22        }
23        ~Event() {
24            cout << "Delete Event class." << endl;
25        }
26 } ;
```

## Event.cpp file (created by Dilrukshi A G T)



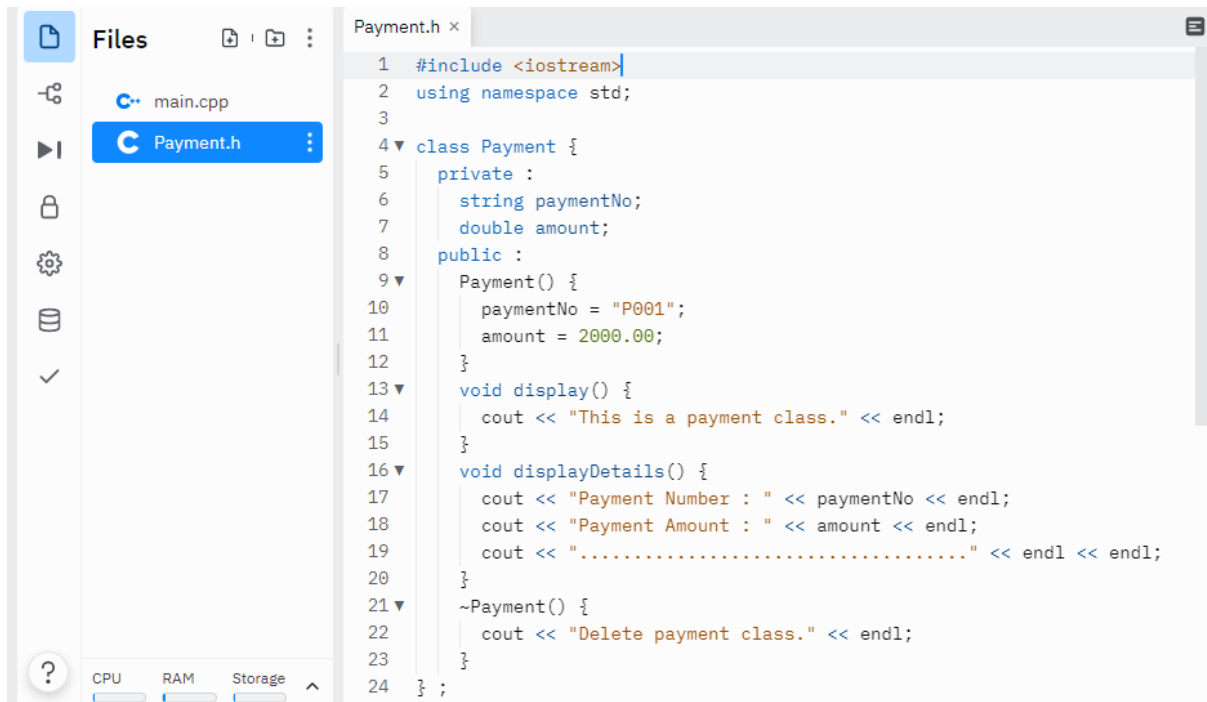
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows two files: main.cpp and Event.h. The main.cpp file is selected and its content is displayed in the code editor. The code includes the Event.h header file and uses the Event class. The console window on the right shows the output of the program.

```
1 #include <iostream>
2 #include "Event.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     Event * e1 = new Event() ;
10
11     // print display() function
12     e1->display();
13
14     //print displayDetails() function
15     e1->displayDetails();
16
17     //calling destructors
18     e1->~Event();
19
20     return 0 ;
21
22 }
```

Console Output:

```
> make -s
> ./main
This is a event class.
Event Number : E001
Event Name : bdays
.....
Delete Event class.
```

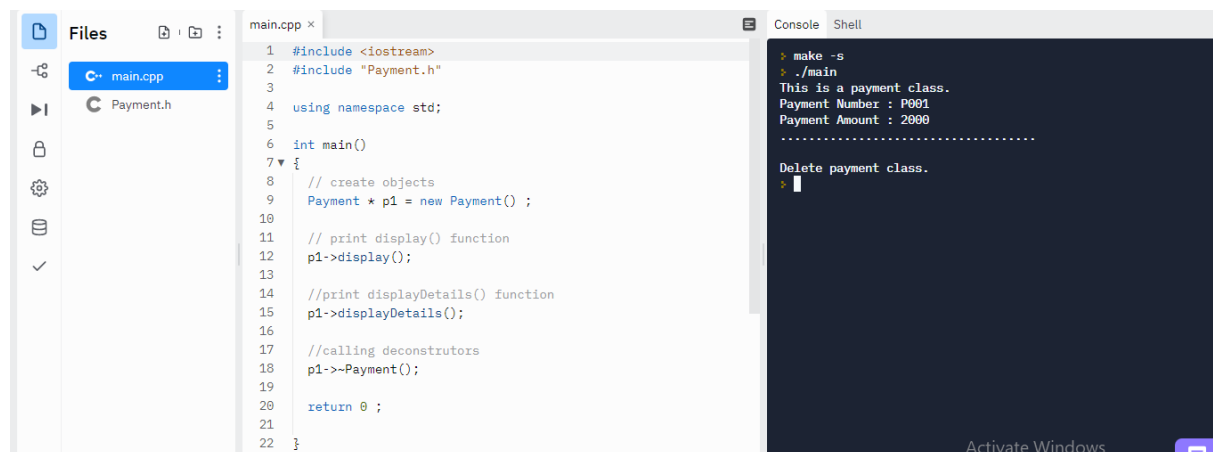
## 08. Payment.h file (created by Kariyapperuma K A D C A)



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'C' with two files: 'main.cpp' and 'Payment.h'. The 'Payment.h' file is selected and its content is displayed in the code editor. The code defines a 'Payment' class with private attributes 'paymentNo' and 'amount', and public methods 'Payment()', 'display()', 'displayDetails()', and '~Payment()'. The 'Payment()' constructor initializes 'paymentNo' to 'P001' and 'amount' to '2000.00'. The 'display()' method prints 'This is a payment class.'. The 'displayDetails()' method prints 'Payment Number : P001' and 'Payment Amount : 2000.00'. The '~Payment()' destructor prints 'Delete payment class.'.

```
1 #include <iostream>
2 using namespace std;
3
4 class Payment {
5     private :
6         string paymentNo;
7         double amount;
8     public :
9         Payment() {
10             paymentNo = "P001";
11             amount = 2000.00;
12         }
13         void display() {
14             cout << "This is a payment class." << endl;
15         }
16         void displayDetails() {
17             cout << "Payment Number : " << paymentNo << endl;
18             cout << "Payment Amount : " << amount << endl;
19             cout << "....." << endl << endl;
20         }
21         ~Payment() {
22             cout << "Delete payment class." << endl;
23         }
24 } ;
```

## Payment.cpp file (created by Kariyapperuma K A D C A)



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'C' with two files: 'main.cpp' and 'Payment.h'. The 'main.cpp' file is selected and its content is displayed in the code editor. The code includes 'Payment.h' and uses the 'Payment' class. The 'main()' function creates a 'Payment' object 'p1', calls 'display()', 'displayDetails()', and '~Payment()' methods, and returns 0. The console on the right shows the output of the program, which matches the expected output from the 'Payment' class methods.

```
1 #include <iostream>
2 #include "Payment.h"
3
4 using namespace std;
5
6 int main()
7 {
8     // create objects
9     Payment * p1 = new Payment() ;
10
11     // print display() function
12     p1->display();
13
14     //print displayDetails() function
15     p1->displayDetails();
16
17     //calling destructors
18     p1->~Payment();
19
20     return 0 ;
21
22 }
```

Console Shell

```
> make -s
> ./main
This is a payment class.
Payment Number : P001
Payment Amount : 2000
.....
Delete payment class.
>
```

# Whole Code of System

main.cpp ×

```
1  #include <iostream>
2  using namespace std;
3
4  //Customer Class
5
6  class Customer {
7  protected :
8      string name;
9      string address;
10     string tel;
11     int age;
12 public :
13     Customer() {
14         name = "Chamath";
15         address = "Mathara";
16         tel = "0755637447";
17         age = 20;
18     } ;
19     void display() {
20         cout << "This is a customer class." << endl ;
21     }
22     void displayDetails() {
23         cout << "Customer Name : " << name << endl;
24         cout << "Customer Address : " << address << endl;
25         cout << "Customer Telephone Number : " << tel << endl;
26         cout << "Customer Age : " << age << endl;
27         cout << "....." << endl << endl;
28     }
29     ~Customer() {
30         cout << "Delete Customer class." << endl;
31     }
32 } ;
33
34 //Event Class
35
36 class Event {
37 private :
38     string eventNo;
39     string eventName;
40 public :
41     Event() {
42         eventNo = "E001";
43         eventName = "bday";
44         Event * eve;
45     }
46     void display() {
```

```

47     cout << "This is a event class." << endl ;
48 }
49 ▼ void displayDetails() {
50     cout << "Event Number : " << eventNo << endl ;
51     cout << "Event Name : " << eventName << endl ;
52     cout << "....." << endl << endl;
53 }
54 ▼ ~Event() {
55     cout << "Delete Event class." << endl;
56 }
57 } ;
58
59 //Payment Class
60
61 ▼ class Payment {
62     private :
63         string paymentNo;
64         double amount;
65     public :
66     Payment() {
67         paymentNo = "P001";
68         amount = 2000.00;
69     }
70     void display() {
71         cout << "This is a payment class." << endl;
72         cout << "....." << endl << endl;
73     }
74     void displayDetails() {
75         cout << "Payment Number : " << paymentNo << endl;
76         cout << "Payment Amount : " << amount << endl;
77         cout << "....." << endl << endl;
78     }
79     ~Payment() {
80         cout << "Delete payment class." << endl;
81     }
82 } ;
83
84 //Bill Class
85
86 ▼ class Bill {
87     private :
88         string billNo;
89         string type;
90         string billName;

```



```

91     double billAmount;
92     Payment * payments[2];
93     //RegisteredCustomer * regcus;
94 public :
95     Bill() {
96         billNo = "B001";
97         type = "Water";
98         billName = "WatterBill";
99         billAmount = 1000.00;
100    }
101    void display() {
102        cout << "This is a bill class." << endl;
103    }
104    void addPayment() {
105        payments[0] = new Payment();
106        payments[1] = new Payment();
107    }
108    void displayDetails() {
109        cout << "Bill Number : " << billNo << endl;
110        cout << "Bill Type : " << type << endl;
111        cout << "Bill Name : " << billName << endl;
112        cout << "Bill Amount : " << billAmount << endl;
113        cout << "....." << endl << endl;
114    }
115    ~Bill() {
116        cout << "Delete Bill class." << endl;
117    }
118 } ;
119
120 //Registered Customer Class
121
122 class RegisteredCustomer : public Customer {
123 private :
124     string customerNo;
125     Bill * bills[1];
126     Event * events[1];
127 public :
128     RegisteredCustomer() {
129         name = "Shashani";
130         address = "Kottawa";
131         tel = "0711108805";
132         age = 20;
133         customerNo = "C001";
134     } ;
135     void display() {

```

```

136     cout << "This is a registered customer class." << endl ;
137 }
138 ▼ void displayDetails() {
139     cout << "Registered Customer Name : " << name << endl;
140     cout << "Registered Customer Address : " << address << endl;
141     cout << "Registered Customer Telephone Number : " << tel << endl;
142     cout << "Registered Customer Age : " << age << endl;
143     cout << "....." << endl << endl;
144 }
145 ▼ ~RegisteredCustomer() {
146     cout << "Delete Registered Customer class." << endl;
147 }
148 } ;
149
150 //Customer Care Person Class
151
152 ▼ class CustomerCarePerson {
153     protected :
154         string empNo;
155         string name;
156         string address;
157         double salary;
158         string tel;
159         int age;
160         Customer * customers[2];
161         RegisteredCustomer * registeredcustomers[2];
162     public :
163     ▼ CustomerCarePerson() {
164         empNo = "E001";
165         name = "Keshala";
166         address = "Kurunegala";
167         tel = "0777523877";
168         age = 20;
169     } ;
170     ▼ void display() {
171         cout << "This is a customer care person class." << endl ;
172     }
173     ▼ void addCustomer() {
174         customers[0] = new Customer();
175         customers[1] = new Customer();
176     }
177     ▼ void addRegisteredCustomer() {
178         registeredcustomers[0] = new RegisteredCustomer();
179         registeredcustomers[1] = new RegisteredCustomer();

```

```

180     }
181 ▼ void displayDetails() {
182     cout << "Customer Care Person Name : " << name << endl;
183     cout << "Customer Care Person Address : " << address << endl;
184     cout << "Customer Care Person Telephone Number : " << tel << endl;
185     cout << "Customer Care Person Age : " << age << endl;
186     cout << "....." << endl << endl;
187 }
188 double calcSalary() {};
189 ▼ ~CustomerCarePerson() {
190     cout << "Delete Customer Care Person class." << endl;
191 }
192 };
193
194 //Web Developer Class
195
196 ▼ class WebDeveloper : public CustomerCarePerson {
197     public :
198     WebDeveloper() {
199         empNo = "E002";
200         name = "Tharushi";
201         address = "Biyagama";
202         tel = "0713616154";
203         age = 22;
204     } ;
205 ▼ void display() {
206     cout << "This is a web developer class." << endl ;
207 }
208 ▼ void displayDetails() {
209     cout << "Web Developer Name : " << name << endl;
210     cout << "Web Developer Address : " << address << endl;
211     cout << "Web Developer Telephone Number : " << tel << endl;
212     cout << "Web Developer Age : " << age << endl;
213     cout << "....." << endl << endl;
214 }
215 double calcSalary() {};
216 ▼ ~WebDeveloper() {
217     cout << "Delete Web Developer class." << endl;
218 }
219 };
220
221 //Administrator Class
222
223 ▼ class Administrator : public WebDeveloper {
224     private :
225     Customer * cus;
226     RegisteredCustomer * regcus;
227     Event * eve;
228     Bill * bil;
229     public :
230 ▼ Administrator() {
231     empNo = "E003";
232     name = "Thevin";
233     address = "Nuwara";
234     tel = "0711555722";
235     age = 22;
236 } ;
237 ▼ void display() {
238     cout << "This is a administrator class." << endl ;

```

```

239     }
240 ▼ void displayDetails() {
241     cout << "Administrator Name : " << name << endl;
242     cout << "Administrator Address : " << address << endl;
243     cout << "Administrator Telephone Number : " << tel << endl;
244     cout << "Administrator Age : " << age << endl;
245     cout << "....." << endl << endl;
246 }
247 double calcSalary() {};
248 ▼ ~Administrator() {
249     cout << "Delete Administrator class." << endl;
250 }
251 } ;
252
253 // This is where the main programe starts
254
255 int main()
256 ▼ {
257     // create objects
258
259     Customer c1;
260     RegisteredCustomer rc1;
261     CustomerCarePerson ccp1;
262     WebDeveloper w1;
263
264     Administrator * a1 = new Administrator() ;
265     Bill * b1 = new Bill() ;
266     Event * e1 = new Event() ;
267     Payment * p1 = new Payment() ;
268
269     // print display() function
270
271     c1.display();
272     rc1.display();
273     ccp1.display();
274     w1.display();
275
276     a1->display();
277     b1->display();
278     e1->display();
279     p1->display();

```

```
280
281 //print displayDetails() function
282
283 c1.displayDetails();
284 rc1.displayDetails();
285 ccp1.displayDetails();
286 w1.displayDetails();
287
288 a1->displayDetails();
289 b1->displayDetails();
290 e1->displayDetails();
291 p1->displayDetails();
292
293 //calling deconstructors
294
295 c1.~Customer();
296 rc1.~RegisteredCustomer();
297 ccp1.~CustomerCarePerson();
298 w1.~WebDeveloper();
299
300 a1->~Administrator();
301 b1->~Bill();
302 e1->~Event();
303 p1->~Payment();
304
305 return 0 ;
306
307
```

# Output

```
Console Shell
> make -s
> ./main
This is a customer class.
This is a registered customer class.
This is a customer care person class.
This is a web developer class.
This is a administrator class.
This is a bill class.
This is a event class.
This is a payment class.
.....

Customer Name : Chamath
Customer Address : Mathara
Customer Telephone Number : 0755637447
Customer Age : 20
.....

Registered Customer Name : Shashani
Registered Customer Address : Kottawa
Registered Customer Telephone Number : 0711108805
Registered Customer Age : 20
.....

Customer Care Person Name : Keshala
Customer Care Person Address : Kurunegala
Customer Care Person Telephone Number : 0777523877
Customer Care Person Age : 20
.....

Web Developer Name : Tharushi
Web Developer Address : Biyagama
Web Developer Telephone Number : 0713616154
Web Developer Age : 22
.....

Administrator Name : Thevin
Administrator Address : Nuwara
Administrator Telephone Number : 0711555722
Administrator Age : 22
.....

Bill Number : B001
Bill Type : Water
Bill Name : WatterBill
Bill Amount : 1000
.....

Event Number : E001
Event Name : bday
.....

Payment Number : P001
Payment Amount : 2000
.....

Delete Customer class.
Delete Registered Customer class.
Delete Customer class.
Delete Customer Care Person class.
Delete Web Developer class.
Delete Customer Care Person class.
Delete Administrator class.
Delete Web Developer class.
Delete Customer Care Person class.
Delete Bill class.
Delete Event class.
Delete payment class.
Delete Web Developer class.
Delete Customer Care Person class.
Delete Registered Customer class.
Delete Customer class.
Delete Customer class.
> |
```

# Contribution of Projects

Student ID	Student Name	Individual Contribution
IT21108822	Abeykoon M W T H B	<ul style="list-style-type: none"> <li>• Write requirement analysis description</li> <li>• Identify classes</li> <li>• Write CRC cards of administrator</li> <li>• Design class diagram of administrator</li> <li>• Code of administrator class</li> <li>• Code of whole class</li> </ul>
IT21044304	Kariyapperuma K A D C A	<ul style="list-style-type: none"> <li>• Write requirement analysis description</li> <li>• Identify classes</li> <li>• Write CRC cards of customer and payment</li> <li>• Design class diagram of customer and payment</li> <li>• Code of customer and payment class</li> <li>• Code of whole class</li> </ul>
IT21224966	Dilrukshi A G T	<ul style="list-style-type: none"> <li>• Write requirement analysis description</li> <li>• Identify classes</li> <li>• Write CRC cards of web developer and event</li> <li>• Design class diagram of web developer and event</li> <li>• Code of web developer and event class</li> <li>• Code of whole class</li> </ul>
IT21237522	Thathsarani H A N N	<ul style="list-style-type: none"> <li>• Write requirement analysis description</li> <li>• Identify classes</li> <li>• Write CRC cards of customer care person</li> <li>• Design class diagram of customer care person</li> <li>• Code of customer care person class</li> <li>• Code of whole class</li> </ul>
IT21227486	Ruhunage S D P R	<ul style="list-style-type: none"> <li>• Write requirement analysis description</li> <li>• Identify classes</li> <li>• Write CRC cards of registered customer and event</li> <li>• Design class diagram of web developer and event</li> <li>• Code of web developer and event class</li> <li>• Code of whole class</li> </ul>