程式設計 HW5 Part3 Coding

108072147 林汶螢

1. Chips and Salsa

```cpp
1.cpp > main()
1    #include <iostream>
2    #include <string>
3    #include <sstream>
4    #include <iomanip>
5    using namespace std;
6    string salsa[]={"mild", "medium", "sweet", "hot","zesty"};
7    int sold[5]={0,0,0,0,0};
8
9    int main(){
10       int jarnum=0,maxsold=0,maxindex=0,leastindex=0,leastsold=0,sum=0;
11       bool startover;
12       for(jarnum=0;jarnum<5;jarnum++){
13           do{
14               startover=0;
15               cout<<"Please enter the jar sold of "<<salsa[jarnum]<<" : ";
16               string input;
17               getline(cin,input);
18               stringstream num(input);
19               if(!(num >> sold[jarnum] )) {
20                   cout<<"You should input nonnegative integer.\n"<<endl;
21                   startover=1;
22               }
23               char ischar;
24               if (num >> ischar){
25                   cout<<"You should input nonnegative integer.\n"<<endl;
26                   startover=1;
27               }
28               if(sold[jarnum]<0){
29                   cout<<"You should input nonnegative integer.\n"<<endl;
30                   startover=1;
31               }
32           leastsold=sold[0];
33           if (sold[jarnum]>=maxsold){ maxsold=sold[jarnum]; maxindex=jarnum; }
34           if (sold[jarnum]<=leastsold){ leastsold=sold[jarnum]; leastindex=jarnum; }
35       }while(startover);
36   }
37   cout<<endl<<"|"<<setfill('=')<<setw(40)<<"|"<<endl;
38   for(jarnum=0;jarnum<5;jarnum++){
39       cout<<"  The jar sold of "<<setfill(' ')<<setw(10)<<salsa[jarnum]<<" is : "<<sold[jarnum]<<endl;
40       sum+=sold[jarnum];
41   }
42   cout<<"  The total sold of salsa is "<<sum<<"."<<endl;
43   cout<<"  The highest selling is "<<salsa[maxindex]<<"."<<endl;
44   cout<<"  The lowest selling is "<<salsa[leastindex]<<"."<<endl;
45   cout<<"|"<<setfill('=')<<setw(40)<<"|"<<endl;
46   return 0;
47 }
48
```

```
PS C:\Users\User\OneDrive\作業\C++HW5> ./1.exe
Please enter the jar sold of mild : 10
Please enter the jar sold of medium : 5
Please enter the jar sold of sweet : -1
You should input nonnegative integer.

Please enter the jar sold of sweet : 6
Please enter the jar sold of hot : 0
Please enter the jar sold of zesty : 7

|======================================|
  The jar sold of       mild is : 10
  The jar sold of     medium is : 5
  The jar sold of      sweet is : 6
  The jar sold of        hot is : 0
  The jar sold of      zesty is : 7
  The total sold of salsa is 28.
  The highest selling is mild.
  The lowest selling is zesty.
|======================================|
```

2. Driver's License Exam

```cpp
1   #include<iostream>
2   #include<sstream>
3   #include<iomanip>
4   using namespace std;
5
6   string rightanswer = "BCCDCDADABABBDDAACCA";
7
8   class TestGrader{
9       private:
10          char answers[20];
11
12      public:
13          void setKey(string answers){
14              for(int i = 0; i < answers.size() && i < 20; i++){
15                  this->answers[i] = answers[i];
16              }
17          }
18          string grade(char stdans[] ){
19              int right=0,wrong=0;
20              bool  wronglist[20],pass;
21              string ret,strright,strwrong,list;
22
23              for(int i = 0; i < 20; i++){
24                  if(this->answers[i] == stdans[i]){
25                      wronglist[i]=0;
26                      right++;
27                  } else {
28                      wronglist[i]=1;
29                      wrong++;
30                  }
```

```cpp
31              }
32              stringstream transfer;
33              transfer << right;
34              transfer >> strright;
35              transfer.clear();
36              transfer << wrong;
37              transfer >> strwrong;
38              transfer.clear();
39
40              if(right>=15) ret += "Pass\n";
41              else          ret += "Fail\n";
42              ret += "Ur right answer is " + strright + ".\n" ;
43              ret += "Ur wrong answer is " + strwrong + ".\n" ;
44              ret +="Wrong number list: ";
45              for(int i=0; i<20; i++){
46                  if (wronglist[i]){
47                      transfer << i;
48                      transfer >> list;
49                      transfer.clear();
50                      ret += list + " ";
51                  }
52              }
53              ret += "\n";
54              return ret;
55          }
56  };
57  int main(int argc, char* argv[]){
58      TestGrader testGrader;
59      testGrader.setKey(rightanswer);
60      string s;
```

```cpp
61        bool flag=0;
62        char answer[20];
63        char isgame;
64        do{
65
66            for(int i=0;i<20;i++){
67                while(true){
68                    cout<<"Please enter the answer for Q"<<i<<" :";
69                    getline(cin,s);
70                    answer[i] = s[0];
71                    if(s.size()>1 || answer[i]<'A' || answer[i]>'D'){
72                        cout<<"Please enter A~D only.\n";
73                    } else {
74                        break;
75                    }
76                }
77            }
78            cout<<testGrader.grade(answer);
79            do{
80                cout<<"Do u want to continue?\n";
81                cout<<"Please enter Y/N only.\n";
82                getline(cin,s);
83                isgame = s[0];
84                if(isgame=='Y') flag =1;
85                else flag=0;
86            }while(s.size()>1 || (isgame != 'Y' && isgame != 'N'));
87        } while (flag);
88
89        return 0;
90    }
```

```
PS C:\Users\User\OneDrive\作業\C++HW5> ./2.exe
Please enter the answer for Q0 :A
Please enter the answer for Q1 :A
Please enter the answer for Q2 :A
Please enter the answer for Q3 :S
Please enter A~D only.
Please enter the answer for Q3 :A
Please enter the answer for Q4 :A
Please enter the answer for Q5 :A
Please enter the answer for Q6 :A
Please enter the answer for Q7 :A
Please enter the answer for Q8 :A
Please enter the answer for Q9 :A
Please enter the answer for Q10 :A
Please enter the answer for Q11 :A
Please enter the answer for Q12 :A
Please enter the answer for Q13 :A
Please enter the answer for Q14 :A
Please enter the answer for Q15 :A
Please enter the answer for Q16 :A
Please enter the answer for Q17 :A
Please enter the answer for Q18 :A
Please enter the answer for Q19 :A
Fail
Ur right answer is 6.
Ur wrong answer is 14.
Wrong number list: 0 1 2 3 4 5 7 9 11 12 13 14 17 18
Do u want to continue?
Please enter Y/N only.
y
Do u want to continue?
Please enter Y/N only.
N
```

3. Lottery

```cpp
#include <iostream>
#include <string>
#include <sstream>
#include <time.h>
#include <cstdlib>
using namespace std;

int main(int argc, char* argv[]){
    int winningDigits[5],player[5];
    string s;
    bool flag=1;
    int matchNum;

    char isgame;
    srand( time(NULL) );
    do{
        for(int i=0;i<5;i++){
            winningDigits[i]=rand() % 10;
        }

        bool cont = 1;
        while(cont){
            cout<<"Please enter ur lucky number:";
            getline(cin,s);
            if(s.size() == 5) {
                cont = 0;
                for(int i = 0; i < 5; i++){
                    if(s[i]<'0' || s[i]>'9') cont = 1;
                    player[i] = s[i] - '0';
                }
            }
            if(cont){
                cout<<"Wrong input, do it again.\n";
            }
        }

        cout << endl;
        cout << "winningDigits:";
        for(int i = 0; i<5; ++i){
            cout << winningDigits[i] << " ";
        }
        cout << endl;

        cout << "player:      ";
        for(int i = 0; i<5; ++i){
            cout << player[i] << " ";
        }
        cout << endl;

        matchNum = 0;
        for(int i = 0; i<5; ++i){
            if(winningDigits[i] == player[i]) matchNum++;
        }

        cout << "You match " << matchNum << " number\n\n";


        cout<<"Do u want to Quit?\n";
        cout<<"Please enter Q only.\n";
        getline(cin,s);
        isgame = s[0];
        if(isgame=='Q') flag =0;
```

```
63              else flag=1;
64        } while (flag);
65        return 0;
66   }
```

```
PS C:\Users\User\OneDrive\作業\C++HW5> ./3.exe
Please enter ur lucky number:11124

winningDigits:2 9 4 6 4
player:      1 1 1 2 4
You match 1 number

Do u want to Quit?
Please enter Q only.
Q
PS C:\Users\User\OneDrive\作業\C++HW5>
```

4. Rainfall Statistics

```cpp
4.cpp > main(int, char * [])
1    #include <iostream>
2    #include <sstream>
3    using namespace std;
4
5    class Stats{
6        private:
7            string city;
8            string monthYear;
9            double rainfall[30];
10           int cnt;
11           int most, least;
12           const string month[12]={  "January", "February", "March",
13                                     "April", "May", "June", "July",
14                                     "August","September","October",
15                                     "November","December"
16                                   };
17       public:
18           Stats(string city, string monthYear){
19               this -> city = city;
20               this -> monthYear = monthYear;
21               for(int i=0; i<30; i++) this -> rainfall[i] = 0;
22               this -> cnt = 0;
23               this -> most = 0;
24               this -> least = 0;
25           }
26           double total(){
27               double sum = 0;
28               for(int i=0; i< this->cnt; i++){
29                   sum += this->rainfall[i];
30               }
31               return sum;
32           }
```

```cpp
        double average(){
            return this->total()/this->cnt;
        }
        double lowest(){
            double L=rainfall[0];
            for(int i=0; i< this->cnt; i++){
                if(this->rainfall[i] < L){
                    least = i;
                    L = this->rainfall[i];
                }
            }
            return L;
        }
        double highest(){
            double M=rainfall[0];
            for(int i=0; i< this->cnt; i++){
                if(this->rainfall[i] > M){
                    most = i;
                    M = this->rainfall[i];
                }
            }
            return M;
        }
        bool storeValue(double rainfall){
            if (rainfall<0 || cnt>30) return false;
            this->rainfall[this->cnt] = rainfall;
            cnt++;
            return true;
        }
        string convert(){
```

```cpp
            stringstream ss;
            int monthnum;

        ss << this->monthYear.substr(4,2);
        ss >> monthnum;
        return this->month[monthnum] + ", " + this->monthYear.substr(0,4);
    }

    string convert(int cnt){
        stringstream ss;
        int monthnum, yearnum;
        string yearstr;

        ss << this->monthYear.substr(4,2);
        ss >> monthnum;
        ss.clear();

        ss << this->monthYear.substr(0,4);
        ss >> yearnum;
        ss.clear();

        yearnum += (monthnum + cnt)/12;
        monthnum = (monthnum + cnt)%12;

        ss << yearnum;
        ss >> yearstr;

        return this->month[monthnum] + ", " + yearstr;
    }

    string displayReport(){
        string ret;
```

```cpp
            ret += this->convert() + " -" + this->convert(cnt) + " Rain Report for " + this->city +"County\n";
            stringstream temp;
            string strtemp;
            temp<< total();
            temp>> strtemp;
            temp.clear();
            ret += "Total rainfall in this period: " + strtemp + " inches\n";
            temp<< average();
            temp>> strtemp;
            temp.clear();
            ret += "Average monthly rainfall: " + strtemp + " inches\n";
            temp<< lowest();
            temp>> strtemp;
            temp.clear();
            ret += "The least rain fell in " + this->convert(least) + " with " + strtemp + " inches\n";
            temp<< highest();
            temp>> strtemp;
            temp.clear();
            ret += "The most rain fell in " + this->convert(most) + " with " + strtemp + " inches\n";
            return ret;
        }
};
int main (int argc, char* argv[]){
    string city,monthYear;
    double rainfall;
    int cont;
    string s;
    char c;

    cout<<"Please enter the city u want to analyze:";
    getline(cin,city);
    cout<<"Please enter the month u started to analyze:";
    cin>> monthYear;
    Stats stats(city, monthYear);

    cont = 1;
    while(cont){
        do{
            cout<<"\nHow's the rainfall in "<<city<<" ?\n";
            cin>>rainfall;
        }while(!stats.storeValue(rainfall));
        getline(cin,s);

        do{
            cout<<"Do u want to continue?\n";
            cout<<"Please enter Y/N only.\n";
            getline(cin,s);
            if(s[0]=='Y') cont =1;
            else cont = 0;
        }while(s.size()>1 || (s[0] != 'Y' && s[0] != 'N'));
    }

    cout<<stats.displayReport();
    return 0;
}
```

```
PS C:\Users\User\OneDrive\作業\C++HW5> ./4.exe
Please enter the city u want to analyze:KH
Please enter the month u started to analyze:200006

How's the rainfall in KH ?
100
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in KH ?
10
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in KH ?
46
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in KH ?
85
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in KH ?
941
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in KH ?
76
Do u want to continue?
Please enter Y/N only.
N
July, 2000 -January, 2001 Rain Report for KHCounty
Total rainfall in this period: 1258 inches
Average monthly rainfall: 209.667 inches
The least rain fell in August, 2000 with 10 inches
The most rain fell in November, 2000 with 941 inches
PS C:\Users\User\OneDrive\作業\C++HW5>
```

5. **Drink Machine Simulator**

```cpp
#include <iostream>
#include <sstream>
#include <iomanip>
using namespace std;
class Vendingmachine{
    private:
    double sum;
    struct {
        string drink;
        double cost;
        int numInmachine;
    } vendingmachine[5];

    double inputMoney(int drinkIndx){
        double money = -1;
        string sin;
        while(money < 0){
            cout<<"Please input money."<<endl;
            getline(cin,sin);
            stringstream ss(sin);
            if(ss >> money ) {
                char ischar;
                if (!(ss >> ischar)){
                    if(money>0){
                        break;
                    }
                }
```

```cpp
27                      }
28                  }
29              cout<<"You should input nonnegative number.\n"<<endl;
30          }
31          return money;
32      }
33      void dailyReport(){
34          cout<<endl
35              <<left<<setw(20)<<"Drink Name"
36              <<left<<setw(22)<<"Number in machine"<<endl;
37          for(int i=0; i<5; i++){
38              cout<<left<<setw(20)<<this->vendingmachine[i].drink
39                  <<left<<setw(22)<<this->vendingmachine[i].numInmachine<<"\n";
40          }
41          cout<<endl;
42          cout<<"!!$ Money made today is : "<<fixed<<setprecision(2)<<this->sum<<"$. $!!"<<er
43
44      }
45      public:
46      Vendingmachine(){
47
48          this->vendingmachine[0].drink = "Cola";
49          this->vendingmachine[1].drink = "Root beer";
50          this->vendingmachine[2].drink = "Orange soda";
51          this->vendingmachine[3].drink = "Grape soda";
52          this->vendingmachine[4].drink = "Bottled water";
53          for(int i=0; i<5; i++){
54              this->vendingmachine[i].cost = 1.00;
55              this->vendingmachine[i].numInmachine = 20;
56          }
57          this->vendingmachine[4].cost = 1.50;
58          this->sum  = 0;
59      }
60      ~Vendingmachine(){
61          dailyReport();
62      }
63      void displayChoices(){
64          cout<<left<<setw(20)<<"Drink Name"
65              <<left<<setw(15)<<"Cost"
66              <<left<<setw(15)<<"Number in machine\n";
67          for(int i=0; i<5; i++){
68              cout<<left<<setw(20)<<this->vendingmachine[i].drink
69                  <<left<<setw(22)<<fixed<<setprecision(2)<<this->vendingmachine[i].cost
70                  <<left<<setw(15)<<this->vendingmachine[i].numInmachine<<"\n";
71          }
72      }
73      void buyDrink(){
74          string sin;
75          bool cont=1;
```

```cpp
76              int i;
77          while(cont){
78              cout<<"Which drink do u want to buy?";
79              getline(cin,sin);
80              for(i=0; i<5; i++){
81                  if (vendingmachine[i].drink == sin){
82                      cont=0;
83                      break;
84                  }
85              }
86              if (cont) cout<<"Please enter drinks on the menu.\n";
87          }
88          double money = inputMoney(i);
89          if( money < vendingmachine[i].cost){
90              cout<<"Your money isn't enough. QQ"<<endl;
91              cout<<"Return money: "<< money<<endl;
92          }
93          else {
94              if( this->vendingmachine[i].numInmachine <= 0 ){
95                  cout<<vendingmachine[i].drink<<" Sold out !!!"<<endl;
96              }
97              else {
98                  cout<<"Here is ur bevarage : "<<vendingmachine[i].drink<<endl;
99                  this->vendingmachine[i].numInmachine --;
100                 this ->sum += this->vendingmachine[i].cost;
101                 cout<<"Return money: "<< money-this->vendingmachine[i].cost<<endl;
102             }
103         }
104         cout<<endl;
105     }
106 };
107 int main(int argv,char* argc[]){
108     bool cont=1;
109     Vendingmachine vendingmachine;
110     string sin;
111     while(true){
112         vendingmachine.displayChoices();
113         do{
114             cout<<"Do u want to buy a drink? (Y/N)";
115             getline(cin,sin);
116             if(sin[0]=='Y') cont =1;
117             else cont = 0;
118         }while(sin.size()>1 || (sin[0] != 'Y' && sin[0] != 'N'));
119         if( cont==0 ) break;
120         vendingmachine.buyDrink();
121     }
122     return 0;
123 }
```

```
PS C:\Users\User\OneDrive\作業\C++HW5> ./6.exe

Bin description        Bin Quantity
 0.Turkey                 0
 1.Chocolate              0
 2.Egg                    0
 3.Yoyo                   0
 4.ChickenBreast          0
 5.Gummy Bear             0
 6.Almond                 0
 7.Cookie                 0
 8.Spaghetti              0

What do U want to do?(Add/Remove/Quit)
Add
Which one do U want to add?(number)
3
How many?
10

What do U want to do?(Add/Remove/Quit)
Remove
Which one do U want to remove?(number)
1
How many?
0
Wrong Quantity or Index

What do U want to do?(Add/Remove/Quit)
Quit

Bin description        Bin Quantity
 0.Turkey                 0
 1.Chocolate              0
 2.Egg                    0
 3.Yoyo                   10
 4.ChickenBreast          0
 5.Gummy Bear             0
 6.Almond                 0
 7.Cookie                 0
 8.Spaghetti              0
PS C:\Users\User\OneDrive\作業\C++HW5> []
```

7.

```cpp
1    #include <iostream>
2    #include <fstream>
3    #include <sstream>
4    #include <iomanip>
5    #include <vector>
6    using namespace std;
7    class TicketManager{
8        private:
9            struct
10            {
11                int price;
12                bool isSold;
13            }seatStructures[15][30];
14
15        public:
16            TicketManager(){
17                fstream seatAvailability, seatPrices;
18                seatAvailability.open("SeatAvailability.dat",ios::in);
19                seatPrices.open("SeatPrices.dat",ios::in);
20                int price;
21                char availability;
22
23                for(int i = 0; i<15; ++i){
24                    seatPrices >> price;
25                    for(int j = 0; j < 30; ++j){
26                        seatAvailability >> availability;
```

```cpp
27                        this->seatStructures[i][j].price = price;
28                        this->seatStructures[i][j].isSold = (availability == '*');
29                    }
30                }
31
32                seatAvailability.close();
33                seatPrices.close();
34            }
35            ~TicketManager(){
36                fstream seatAvailability;
37                seatAvailability.open("SeatAvailability.dat",ios::out);
38
39                for(int i = 0; i<15; ++i){
40                    for(int j = 0; j < 30; ++j){
41                        if(this->seatStructures[i][j].isSold){
42                            seatAvailability<<'*';
43                        } else {
44                            seatAvailability<<'#';
45                        }
46                    }
47                    seatAvailability<<endl;
48                }
49
50                seatAvailability.close();
51            }
52
53            string getSeatChart(){
```

```cpp
53          string getSeatChart(){
54              stringstream ss;
55              string temp;
56
57              string ret = "";
58              ret = "                          Seats                \n";
59              ret += "          123456789012345678901234567890\n";
60
61              for(int i = 0; i < 15; ++i){
62                  ss<<i;
63                  ss>>temp;
64                  ss.clear();
65                  ret += "Row ";
66                  if(i<10) ret += " " + temp + "  ";
67                  else     ret += temp + "  ";
68                  for(int j = 0; j < 30; ++j){
69                      if(this->seatStructures[i][j].isSold){
70                          ret += "*";
71                      } else {
72                          ret += "#";
73                      }
74                  }
75                  ret += "\n";
76              }
77
78              return ret;
79          }
80
81      string tickRequest(vector<pair<int,int> > seats){
82          stringstream ss;
83          string stemp;
84
85
86          bool errorLocation = false;
87          int cost = 0;
88          string ret = "Your Seats:";
89          for(int i = 0; i<seats.size(); ++i){
90              if( (seats[i].first < 0 || seats[i].first >= 15 || seats[i].second < 0 || seats[i].second >=30 ) ||
91                  (this->seatStructures[seats[i].first][seats[i].second].isSold) ){
92                  errorLocation = true;
93              } else {
94                  ss << seats[i].first;
95                  ss >> stemp;
96                  ss.clear();
97                  ret += "(" + stemp;
98
99                  ss << seats[i].second;
100                 ss >> stemp;
101                 ss.clear();
102                 ret += "," + stemp + ") ";
103
104                 cost += this->seatStructures[seats[i].first][seats[i].second].price;
105             }
```

```cpp
            }
        }

        if(errorLocation){
            return "Error LOcation\n";
        } else{
            ss << cost;
            ss >> stemp;
            ss.clear();
            ret += "\nTotal Cost:" + stemp + "\n";
            return ret;
        }


        return "";
    }

    string tickPurchase(vector<pair<int,int> > seats, int pay){
        stringstream ss;
        string stemp;

        int cost = 0;
        string ret = "Your Seats:";
        for(int i = 0; i<seats.size(); ++i){
            ss << seats[i].first;
            ss >> stemp;
            ss.clear();
            ret += "(" + stemp;

            ss << seats[i].second;
            ss >> stemp;
            ss.clear();
            ret += "," + stemp + ") ";

            cost += this->seatStructures[seats[i].first][seats[i].second].price;
        }

        if(cost > pay){
            return "Insufficient pay!!!!!\n";
        } else{
            for(int i = 0; i<seats.size(); ++i) {
                this->seatStructures[seats[i].first][seats[i].second].isSold = true;
            }
            ret += "\nSuccessfully paid.\n";
            return ret;
        }
    }

    string report(){
        stringstream ss;
        string temp;

        string ret = "";
```

```cpp
            int soldNum = 0;
            int cllectMoney = 0;

            for(int i = 0; i < 15; ++i){
                for(int j = 0; j < 30; ++j){
                    if(this->seatStructures[i][j].isSold){
                        ++soldNum;
                        cllectMoney += this->seatStructures[i][j].price;
                    }
                }
            }

            ret = "Number of sold seat:     ";
            ss << soldNum;
            ss >> temp;
            ss.clear();
            ret += temp;
            ret += "\n";

            ret = "Number of available seat:";
            ss << ((15*30)-soldNum);
            ss >> temp;
            ss.clear();
            ret += temp;
            ret += "\n";

            ret = "Collected Money:          ";
            ss << cllectMoney;
            ss >> temp;
            ss.clear();
            ret += temp;
            ret += "\n";

            return ret;
        }

};
int main(int argc,char* argv[]){

    TicketManager ticketManager;

    bool cont = true;
    while(cont){
        int select = 0;

        cout<<endl
            <<setfill('=')<<setw(15)<<"Option"<<setw(15)<<"\n"
            <<"  0 :    Display the seating chart "<<endl
```

```cpp
207                    <<"  1 :      Request tickets            "<<endl
208                    <<"  2 :      print sales report         "<<endl
209                    <<"  3 :      EXIT                        "<<endl;
210                cin>>select;
211                if(select == 0){
212                    cout<<ticketManager.getSeatChart();
213                } else if(select == 1){
214                    vector<pair<int,int> > seats;
215                    bool cont = true;
216                    while(cont){
217                        int setI,setJ;
218                        char c;
219
220                        cout<<"Which seat do U want to buy?(example: 10 5)\n";
221                        cout<<"Follow the format plz\n";
222                        cin>>setI;
223                        cin>>setJ;
224                        pair<int, int> seat(setI,setJ);
225
226                        seats.push_back(seat);
227
228                        cout<<"More seat?(Y/N)\n";
229                        cin>>c;
230                        if(c == 'Y') cont = true;
231                        else cont = false;
232                    }
```

```cpp
233                }
234                string retMsg;
235                retMsg = ticketManager.tickRequest(seats);
236                cout << retMsg;
237                if(retMsg != "Error LOcation\n"){
238                    char purchase;
239                    cout << "Are you sure to purchase?(Y/N) \n";
240                    cin>>purchase;
241                    if(purchase == 'Y'){
242                        int pay;
243                        cout << "Pay Now(input money)!!!!!\n";
244                        cin >> pay;
245                        cout << ticketManager.tickPurchase(seats, pay);
246                    } else {
247                        cout << "Ok! Fine!" << endl;
248                    }
249                }
250
251            } else if( select == 2) {
252                cout<<ticketManager.report();
253            } else if( select == 3){
254                cont = false;
255            } else{
256                cout<<"Wrong Input!"<<endl;
257            }
258
```

```
252                     cout<<ticketManager.report();
253             } else if( select == 3){
254                 cont = false;
255             } else{
256                 cout<<"Wrong Input!"<<endl;
257             }
258
259
260         }
261
262
263     return 0;
264 }
```

```
PS D:\Wen homework> ./HW5-7.exe

=========Option===============
  0 :    Display the seating chart
  1 :    Request tickets
  2 :    print sales report
  3 :    EXIT
0
                    Seats
        123456789012345678901234567890
Row  0  ##############################
Row  1  ##############################
Row  2  ##############################
Row  3  ########*#####################
Row  4  ##############################
Row  5  ##############################
Row  6  ##############################
Row  7  ##############################
Row  8  ##############################
Row  9  #######*######################
Row 10  #####*########################
Row 11  ##############################
Row 12  ##############################
Row 13  ##############################
Row 14  ##############################

=========Option===============
  0 :    Display the seating chart
  1 :    Request tickets
  2 :    print sales report
  3 :    EXIT
```