

Object Oriented Programming 11206

First Exam, Fall 2016/2017

November 5, 2016

Instructors:

Dr. Raad Al-Qassas

Dr. Nailah Al-Madi

Dr. Anas Abu Taleb

Student Name: _____

Student Number: _____

Section Number /Doctor: _____

Question	Points	Score
1	48	
2	32	
3	20	
Total	100	

Question 1 (48 points)

Write a C++ class to represent an Elevator (مصعد), which has the following:

- A. Four data members: int **MaxFloor** (أعلى طابق) , int **capacity** (max number of riders/ سعة) , int **currentFloor** (الطابق الحالي) , int **numRiders** (current number of riders راكبين) (4 points)
- B. A default constructor that initializes all data member to 0. (3 points)
- C. An initializer constructor that receives values for **MaxFloor** in the building and **Capacity**. Also sets the **currentFloor** to 0 and **numRiders** to 0. (3 points)
- D. Define setter functions for **MaxFloor** and **capacity** data members. (4 points)
- E. Define getter functions for **numRiders** and **CurrentFloor**. (4 points)
- F. **print()** that prints the current state of the elevator (**currentFloor** and **numRiders**). (2 points)
- G. **boolean IsFull()** which checks if the elevator is full or not, based on the **numRiders** and the **Capacity**. (6 points)
- H. **addRiders (int numPersons)** which adds the **numPersons** into the elevator, make sure that the elevator has a maximum capacity, if the current **numRiders** with **numPersons** exceeds the **capacity** of the elevator don't change the **numRiders** and print a message "Beeb Beeb" (7 points)
- I. **void goUpDown(int floors)** which changes the **currentFloor** of the elevator based on the floors parameter received, such as goUpDown(2) will move the elevator up by 2 floors, goUpDown(-1) will move the elevator down one floor. Keep in mind that the elevator has MaxFloor and the minimum floor is 0. (6 points)
- J. **bool isLarger(Elevator &e)** which returns true if the current elevator has larger capacity than the received elevator e object. (6 points)

```
#include <iostream>
using namespace std;
class Elevator {
public:
    Elevator() {
        MaxFloor = Capacity = currentFloor = numRiders = 0;
    }
    Elevator(int a, int b) {
        MaxFloor = (a>0)?a:0;
        Capacity = (b>0)?b:0;
        currentFloor = numRiders = 0;
    }
    ~Elevator() {
    }
    void setMaxFloor(int a) {
        MaxFloor = (a>0)?a:0;
    }
    void setCapacity(int b) {
        Capacity = (b>0)?b:0;
    }
    int getMaxFloor() {
        return MaxFloor;
    }
}
```

```
int getCapacity() {
    return Capacity;
}
int getCurrentFloor() {
    return currentFloor;
}
void print() {
    cout << "currentFloor: " << currentFloor << endl
        << "numRiders: " << numRiders << endl;
}
bool IsFull() {
    return numRiders == Capacity;
}
void addRiders(int numPeople) {
    numPeople = (numPeople >= 0) ? numPeople: 0;
    if ((numRiders + numPeople) > Capacity)
        cout << "Beeb Beeb \n";
    else
        numRiders += numPeople;
}
```

```

void goUpDpwn(int floors) {
    if ( currentFloor + floors > MaxFloor)
        currentFloor = MaxFloor;
    else if ( currentFloor + floors < 0)
        currentFloor = 0;
    else
        currentFloor += floors;
}

```

```

bool isLarger(Elevator& E) {
    return Capacity > E.Capacity;
}
private:
    int MaxFloor;
    int Capacity;
    int currentFloor;
    int numRiders;
};

```

Question 2 (32 points)

Write a main function that uses the Elevator class done in the previous question. Assume that you have implemented the Elevator class correctly with all functions given. Also, ***assume that you have setters and getters functions for all member variables.*** In the main function, do the following:

- Define two Elevator objects **E1** and **E2** where **E1** uses the default constructor and **E2** uses the initializer constructor. Initialize E2 with 5 for the MaxFloor and 10 for the capacity. (4 points)
- Define an array of 10 Elevator objects and call it **MyMall**. Read the MaxFloor and capacity of each Elevator from user input. (8 points)
- Print the information of the elevator objects which are in the highest floor in array **MyMall**. Note that the elevator might not be the one with the highest MaxFloor. (10 points)
- Print the information of the elevator that has the largest capacity, make use of **isLarger** member functions. (10 points)

```

int main()
{
    Elevator E1;
    Elevator E2(5,10);
    Elevator myMall[10];
    int cap, floor;
    for(int i=0;i<10;i++)
    {
        cin>>cap>>floor;
        myMall[i].setMaxFloor(floor);
        myMall[i].setCapacity(cap);
    }
    int max = myMall[0].getCurrentFloor();
    int index=0;
    for(int i=1;i<10;i++) {
        if(myMall[i].getCurrentFloor()>max){
            max= myMall[i].getCurrentFloor();
            index=i;
        }
    }
}

```

```

myMall[index].print();
cout<< myMall[index].getMaxFloor();
cout<< myMall[index].getCapacity();

int largest=0;
for(int i=1;i<10;i++) {
    if(myMall[i].isLarger(myMall[largest])){
        largest =i;
    }
}
cout<< myMall[largest].getMaxFloor();
cout<< myMall[largest].getCapacity();
myMall[largest].print();
return 0;
}

```

Question 3 (20 points): Trace the following program and write the **generated output in the box below only?**

<pre> class Data2 { private: int h, k; public: Data2(int v = 1); int GetVals(); int SetVals(int value); Data2 GetObj(int value); void UseObj(Data2 &obj); }; Data2::Data2(int v) { h = 3 * v; k = 4 + v; cout << h << "- " << k << endl; } int Data2::GetVals() { return h + k; } int Data2::SetVals(int value) { h = value; k = value + 2; return h + k; } Data2 Data2::GetObj(int value){ Data2 temp(value); h++; k += value; temp.h = h + 2; return temp; } void Data2::UseObj(Data2 &obj){ h += obj.h; k = obj.k * 2; } </pre>	<pre> int main() { Data2 arr[2]; int res = 1; for (int i = 0; i < 2; i++) { res = arr[i].SetVals(i + res); cout << arr[i].GetVals() << endl; } Data2 t=arr[0].GetObj(4); t.SetVals(2); cout << t.GetVals() << endl; arr[1].UseObj(t); cout << arr[1].GetVals() << endl; return 0; } </pre> <p>Output</p> <pre> 3- 5 3- 5 4 12 12- 8 6 15 </pre>
--	---