FAST, LHR(Mid-term Exam OOP 3A)

Mr. Zaeem Yousaf

Course Instructor: Ms. Mamoona Majid

Student Name	Roll No	Section	Q1 Marks	Q2 Marks	Total Marks	Version
			20	20	40	V2

NOTE:

You are given a main code for both questions

- Copy main code and implement

Q1 (marks: 20)

Implement a Cinema Class to run the following main

A Cricket Stadium has the following attributes

- seating capacity
- spectator's list

Following are Member functions

- Parametrized Constructor
- Copy Constructor: 3 marks
- addSpectator(string name, int id): 1 marks
- removeSpectator(int id): 1 marks
- print
- destructor: 5 marks
- Code execution: 10 marks

```
CricketStadium *c1 = new CricketStadium(20); // cinema with 10 seats capacity
// you can only add 10 spectators
// if you try to add more member than capacity
// it will call the following function: exit(1)
for(int i=0; i < 20; i++){
    c1->addSpectator("babar azam",i);
}

// display movie list
// display spectator's list
c1->print()

// make sure you cannot add more than capacity
c1->addSpectator("babar azam",20);
c1->print()

// make sure your copy constructor is ok
CricketStadium c2 = *c1;
// make sure c2 has same contents
delete(c1);
c1 = NULL;

c2.print()
// make sure to remove all spectators
// if id does not exists, it shoud do nothing
for(int i=0; i < 21; i++){
    c2.removeSpectator(i);
}

// make sure there is no spectators
// it should print nothing
c2.print()
```

Q2 (marks: 20)

Zaeem wants to write an Namikaza database' for the following scenario

Marking Scheme

- printAnatoDb: 10 marks
- allocate memory for matrix: 5 marks
- allocate memory for anato database: 5 marks

```c
// write a function to print this database
void printAnatoDb(int *** anato_database, int size){
        // make sure to print all matrix
        // separated by new line
}

int main(int argc, char *argv[])
{
    int odd1[] = {1,3,5,1,3,5,1,3,5};
    int eve1[] = {2,4,6,2,4,6,2,4,6};

    int odd2[] = {13,15,17,19,21,23};
    int eve2[] = {14,16,18,20,22,24};

    // allocate memory for matrix1
    // so that we can store odd1 and even1 to matrix1
    // use for loop to write data to matrix
    int ** matrix1;
    // allocate memory for matrix2
    // so that we can store odd2 and even2 to matrix2
    // use for loop to write data to matrix
    int ** matrix2;

    // now store both of the matrix1 and matrix2 to anato_database
    // allocate memory fo anato_database so that
    // we can store matrix1 and matrix2 to this database
    int *** anato_database;

    // now call print_anato_db(anato_database,2)
     // it will print matrix1, matrix2
    print_anato_db(anato_database,2);

    return 0;
}
```