Object Oriented Programming
Instructor: Natalia Chaudhry

Assignment 3: Space firing game using inheritance and polymorphism

Deadline to submit the assignment is Tuesday 1st December, 2020 till 11:59 pm

Submission instructions **Marks: 30**

1. Submit your file in a .zip or .rar format. Name your file with your Roll number (e.g. BSEF19M009.zip or BSEF19M009.rar)
2. Make a separate folder for each task of assignment (e.g. task1, task2…). All the headers and .cpp files must be in respective folder with a screenshot of the output of a program of the respective task.
3. Every Cpp file must contain your name and roll number(In comments) at the top of each program
4. Don't enclose your executable code in comments otherwise it will not be evaluated.

Instructions: (MUST READ)

- **No compensation or makeup assignment.**
- **Don't discuss with peers. Changing variable names/ changing for to while loop will not help you in hiding cheating attempt!**
- **You are not allowed to ask TA to verify/ prove your cheating case! Any such complaint from TAs will result in serious consequences. Don't expect any positive response from TAs in such regard.**
- **Cheating cases will result in deduction in sessionals.**
- **You are not allowed to consult Internet. Plagiarism cases will be strictly dealt.**
- **Queries are not allowed. Do whatever you are able to understand**

General instructions for all tasks: (marks will be deducted if the instructions are violated)

Note: All the programs should be implemented using class. You can take input in main() function and then call appropriate methods/ member functions of a designed class to set and get values. YOU MUST CREATE A SEPARATE CPP AND HEADER FILE(S) FOR CLASS DECLARATION AND DEFINITION.

• The attributes of class should be declared as **private** and member functions as **public**.

• All the member functions (expect constructor) should be declared inside the class and defined outside the class.

• You should not initialize the attributes while declaring them in class. The values should be assigned using member functions only. E.g. you cannot declare like:

```
Class Person
{
        Private:
        int age=25;
}
```

• The values should be initialized using a constructor. There must be a constructor in your defined class.

- All inputs should be taken in *main()* and all the final results should also be reported/ displayed in the main function.
- All the logic should be implemented in class' member functions. Main() should only input and output relevant values by calling relevant functions of the class.

Space Firing Game

You are required to implement a basic space firing game in C++ using the concepts of inheritance and polymorphism.

Description

The game starts with a welcome screen displaying a rocket-like object and text START written underneath it. The user is required to press 's' to start the game. As soon as the game starts, enemy spaceship objects start appearing from different locations from the top of the screen. The user's player appears as a rocket object at the bottom of the screen. It can be moved left and right using the left and right arrow keys. The player can fire at the enemy spaceships using 'f' keys. The fire object should be visible on the screen. The enemy spaceships also fire back at random intervals of time. The player gets 5 points if it destroys a small spaceship and 10 if it destroys a big spaceship. When the score of the player reaches a certain threshold (say 100), the game stops and the player wins. If the player gets hit by an enemy spaceship three time consecutively, it is destroyed and the game restarts from the beginning. The player gets three lives to win the game. If he does not win in these chances, the game is lost.

Class Descriptions

You can create the following classes (add more if you require them).

1. Spaceship class – This can be an abstract class having one pure virtual method draw() and two virtual methods fire() and move(). The move() function only moves the ship vertically from top to down. The data members of the class can be the location of the spaceship on screen given by (xloc,yloc) coordinates.

2. Two classes can be derived from spaceship class, namely MartianShip and StarShip. MartianShip is bigger in size and more powerful whereas the StarShip is smaller. The MartianShip is destroyed when the player fire five consecutive fireballs at it. The StarShip is destroyed with only three fireballs. Also, they are drawn on screen differently.

3. Rocket class that represents the player. It can store information about number of lives of player and its location (xloc,yloc). Draw() and move() functions can be defined for this class also. The player can move both horizontally and and vertically. A function fire() is needed that fires a fireball at the enemy spaceship.

4. FireBall class. This is contained in all Spaceship and Rocket classes as an object array. Each time the player or enemy fires a fireball the size of the array is decremented.

Score should be kept as a static variable in the Spaceship class and updated whenever a spaceship is destroyed.

Write a main function that demonstrates the working of this game. You can implement this in the text mode using different shapes to display ships and fireballs.