```
public class Game implements Comparable<Game>
{
    private static int globalId=1;
    private int gameId;
    private int rating;
    private String name;


    public Game(String name, int rating){
        this.name=name;
        this.rating=rating;
        this.gameId=globalId++;
    }

    //getter
    public int getId(){
        return gameId;
    }
     public String getName(){
         return name;
     }

    public int compareTo(Game otherGame){
        return this.gameId-otherGame.gameId;
    }



    public String toString(){
            return "GameId : "+gameId+", name : "+this.name+", rating : "+rating;
    }
}
```

```
//
import java.util.*;
class GameManagement{
    static List<Game> gameList = new ArrayList<>();
    static Map<Integer,Game> gameMap = new HashMap<>();     //give key value to
user, take detail of class


    public static void addGame(Game gm){
        gameList.add(gm);                                   //add name
        gameMap.put(gm.getId(),gm);
        System.out.println("added game with Id ."+gm.getId());
```

```
    }

    public static void removeGame(int gameId){
        Game game = gameMap.get(gameId);
        gameList.remove(game);
        gameMap.remove(game.getId());
        System.out.println("game deleted by ID : "+gameId);
    }

    public static Game searchGameById(int gameId){
        Game game = gameMap.get(gameId);
        return game;

    }

    public static List<Game> viewAllGameSortById(){
        Collections.sort(gameList);
        return gameList;
    }

    public static List<Game> viewAllGameSortByName(){
        gameList.sort((g1,g2)->g1.getName().compareTo(g2.getName()));  //if string
compareTo
        return gameList;
    }

    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int n=0;

        do{
            System.out.println("1.Add game, \n2.Remove game, \n3.SearchGameById,
\n4.View all Game, \n5.Sort by name, \nEnter 0 to EXIT");
            n=sc.nextInt();
            sc.nextLine();
            switch(n){
                case 1:
                System.out.println("enter game name.");
                String gname= sc.nextLine();
                System.out.println("enter game rating.");
                int grating = sc.nextInt();
                sc.nextLine();
                if(grating>5 || grating<=0){
                    throw new InvalideRatingException("rating is invalid, please
enter between 1 to 5 d");
                }
                Game gm=new Game(gname,grating);        //constructor
                addGame(gm);
                break;
                case 2:
                System.out.println("enter game Id.");
                int gameId = sc.nextInt();
                removeGame(gameId);
                break;
```

```
                case 3:
                System.out.println("enter game Id.");
                gameId = sc.nextInt();
                searchGameById(gameId);
                break;
                case 4:
                System.out.println("Sorted List id: ");
                List<Game> sortListById =  viewAllGameSortById();
                sortListById.forEach(s->System.out.println(s));
                break;
                case 5:
                System.out.println("Sorted List by name : ");
                List<Game> sortListByName =  viewAllGameSortByName();
                sortListByName.forEach(s->System.out.println(s));


                break;
                default:
                    n=0;
            }
        }while(n>0);


    }

}
```

```
public class InvalideRatingException extends RuntimeException{
    public InvalideRatingException(String msg){
        super(msg);
    }
}
```