

Arrays of Objects

1. Describe the output of the following program that uses data from the James Bond[®] movie series. Ratings for each of the movies are averages compiled from many movie critics by www.all-reviews.com. The data.txt file can be found after problem #3 on page 4.

```
import chn.util.*;
import apcslib.*;

public class Collection
{
    private Movie[] movieList;

    public Collection(String fileName)
    {
        loadData(fileName);
    }

    private void loadData(String fileName)
    {
        String movieTitle;
        String bondName;
        int yearReleased;
        double movieRating;
        int lengthHours;
        int lengthMinutes;

        FileInputStream inFile = new FileInputStream(fileName);

        int numReleases = inFile.readInt();
        movieList = new Movie[numReleases];

        for (int i = 0; i < numReleases; i++)
        {
            movieTitle = inFile.readLine();
            bondName = inFile.readLine();
            yearReleased = inFile.readInt();
            movieRating = inFile.readDouble();
            lengthHours = inFile.readInt();
            lengthMinutes = inFile.readInt();

            movieList[i] = new Movie(movieTitle, bondName, yearReleased,
                                     movieRating, lengthHours, lengthMinutes);
        }
    }
}
```

```

public void displayInfo()
{
    System.out.print(Format.center("Film Title",34));
    System.out.print(Format.center("Bond Actor",16));
    System.out.print(Format.center("Year",6));
    System.out.print(Format.center("Rating",7));
    System.out.print(Format.right("Film Length",16));
    System.out.println();
    System.out.println();
    for (int i = movieList.length - 1; i >= 0 ; i--)
    {
        String output = movieList[i].toString();
        System.out.println(output);
    }
}

public void Sort()
{
    quickSort(movieList,0,movieList.length - 1);
}

private void quickSort(Movie[] list, int first, int last)
{
    int g = first;
    int h = last;
    int midIndex;
    Movie dividingValue;

    midIndex = (first + last) / 2;
    dividingValue = list[midIndex];
    do
    {
        while (list[g].compareTo(dividingValue) < 0)
        {
            g++;
        }
        while (list[h].compareTo(dividingValue) > 0)
        {
            h--;
        }
        if (g <= h)
        {
            // swap g and h
            Movie temp = list[g];
            list[g] = list[h];
            list[h] = temp;
            g++;
            h--;
        }
    } while (g < h);
    if (h > first)
    {
        quickSort(list, first, h);
    }
    if (g < last)
    {
        quickSort(list, g, last);
    }
}

```

```

public static void main(String[] args)
{
    Collection seriesData = new Collection("bond.txt");
    seriesData.Sort();
    seriesData.displayInfo();
}

}
//----- End of Collection class -----//

public class Movie implements Comparable
{
    private String myTitle;      // title of Bond film
    private String myBondActor;  // name of actor of portrayed James Bond
    private int myYear;          // year film was released
    private double myFilmRating; // from all-reviews.com
    private int myLengthHours;   // hours (truncated) portion of film length
    private int myLengthMinutes; // minutes beyond truncated hours

    public Movie(String title, String name, int yr, double rating, int hrs, int min)
    {
        myTitle = title;
        myBondActor = name;
        myYear = yr;
        myFilmRating = rating;
        myLengthHours = hrs;
        myLengthMinutes = min;
    }

    public String getTitle()
    {
        return myTitle;
    }

    public String getBondActor()
    {
        return myBondActor;
    }

    public int getYearFilmReleased()
    {
        return myYear;
    }

    public double getFilmRating()
    {
        return myFilmRating;
    }

    public int getFilmHrs()
    {
        return myLengthHours;
    }

    public int getFilmMin()
    {
        return myLengthMinutes;
    }
}

```

```

public int compareTo(Object other)
{
    return (int) (myFilmRating*10) - (int) (((Movie)other).myFilmRating)*10);
}

public String toString()
{
    return (Format.left(myTitle,34) + Format.left(myBondActor,16)
        + Format.center(myYear,6) + Format.center(myFilmRating,7,1)
        + "      " + myLengthHours + " hr " + myLengthMinutes + " min");
}
}

```

2. Revise the *Collection* class to calculate and display (to the screen) the average rating AND average length for all James Bond movies. Your enhancements should include methods called *displayAveRating* and *displayAveMinutes* in this class. Your additional output should look as follows:

The average rating for a James Bond Movie is 2.41 out of a possible 4.0

The average length for a James Bond Movie is 2 hrs and 6.2 minutes

3. Revise the program to sort the movies by year of release.

data.txt to be used for this program (continued on the next page):

```

22
Thunderball
Sean Connery
1965 3 2 10
Live and Let Die
Roger Moore
1973 2 2 1
Goldeneye
Pierce Bronson
1995 2.5 2 10
Dr. No
Sean Connery
1962 2.5 1 50
You Only Live Twice
Sean Connery
1967 2.5 1 57
Diamonds Are Forever
Sean Connery
1971 3 2 00
From Russia With Love
Sean Connery
1963 3 1 55
Never Say Never Again

```

Sean Connery
1983 1.5 2 14
Tomorrow Never Dies
Pierce Bronson
1997 1.5 1 57
Die Another Day
Pierce Bronson
2002 2.5 2 12
Goldfinger
Sean Connery
1964 4 1 50
On Her Majesty's Secret Service
George Lazenby
1969 2 2 22
Casino Royal
David Niven
1967 2 2 11
The World is Not Enough
Pierce Bronson
1999 2 2 8
A View To A Kill
Roger Moore
1985 2 2 11
For Your Eyes Only
Roger Moore
1981 2.5 2 8
The Man With The Golden Gun
Roger Moore
1974 2 2 05
The Living Daylights
Timothy Dalton
1987 2 2 11
Moonraker
Roger Moore
1979 3 2 16
Licence To Kill
Timothy Dalton
1989 2 2 13
The Spy Who Loved Me
Roger Moore
1977 3 2 05
Octopussy
Roger Moore
1983 2.5 2 11