

## Handing In

“You are expected to hand in your code for the solution of **EXERCISE 2 ONLY** by *Thursday 11th of March, 12:00pm*. Make sure that your code is properly indented and commented and that your name and ugxxxx number appears on your paper. Exercises 1 and 3 will **NOT** be marked, although you are welcomed to hand them in as well; you will get *comments* on those too.”

### Exercise 2: Program Code

```
/**
 *
 * Lab 12, Exercise 2
 *
 * Written by Gareth Evans
 *
 * 25/Feb/99
 */

import java.util.* ;
import java.bangor.*;

public class VideoShop
{
    /**
     * main
     */

    public static void main(String arg[]) throws Exception
    {
        // Define the Variables

        int[] Videos;
        Videos = new int[20];

        int Choice;
        int HireVideo;
        int HireCustomer;
        int ReturnVideo;
        int ReturnCustomer;
        int CheckVideo;
        int CheckCustomer;

        // Note: The Videos array will store the number of videos in stock

        int[] Customers;
        Customers = new int[50];

        // Note: The Customers Array will store which video the customer
        //         Currently has ( the value -1 = no video)

        // Set the Variable Values

        for (int i = 0; i < 10; i++)
        { Videos[i] = 4; } // We have 4 copies of Videos 0 to 9

        for (int j = 10; j < 20; j++)
        { Videos[j] = 6; } // We have 6 copies of Videos 10 to 19

        for (int k = 0; k < 50; k++)
        { Customers[k] = -1; } // No videos are hired out at the start
    }
}
```

```

// MENU SYSTEM

do
{
    System.out.println();
    System.out.println("MAIN MENU - PLEASE ENTER YOUR CHOICE");
    System.out.println("-----");
    System.out.println();
    System.out.println("1. Video Hire");
    System.out.println("2. Video Return");
    System.out.println("3: Tapes in Stock");
    System.out.println("4. Customer Information");
    System.out.println();
    System.out.println("5. Exit");
    System.out.println();
    Choice = BasicIo.readInteger();

    // 1. VIDEO HIRE

    if (Choice==1)
    {
        System.out.println("Which video do you want to hire?");
        System.out.println("Enter a number between 0 and 19");
        HireVideo = BasicIo.readInteger();

        if (Videos[HireVideo]>0) // if the video is in stock
        {
            System.out.println();
            System.out.println("Which customer wants to hire this video?");
            System.out.println("Enter a number between 0 and 49");
            HireCustomer = BasicIo.readInteger();

            if ((Customers[HireCustomer]) == (-1)) // if the customer has not
                // already hired a video
            {
                Customers[HireCustomer] = HireVideo;
                Videos[HireVideo] = (Videos[HireVideo] - 1); // reduce stock level
            }
            else // i.e. customer has a video already
            {
                System.out.println();
                System.out.println("The Customer selected has already hired a
                    video");
                System.out.println("Please return this video before attempting to
                    hire another video");
            }
        }
        else // i.e. number in stock = 0
        {
            System.out.println();
            System.out.println("There are no copies of this video in stock.
                Sorry");
        }
    }

    // 2. VIDEO RETURN

    if (Choice==2)
    {
        System.out.println();
        System.out.println("Which Customer is Returning a video?");
        ReturnCustomer = BasicIo.readInteger();
        if (Customers[ReturnCustomer]>-1) // i.e. customer HAS hired a video
        {
            ReturnVideo = Customers[ReturnCustomer];
            Videos[ReturnVideo] = (Videos[ReturnVideo] + 1); // increase stock
            Customers[ReturnCustomer] = -1; // flag customer as having no video
        }
        else // i.e. Customers[ReturnCustomer] = -1
        {
            System.out.println();
            System.out.println("The selected customer has no videos to
                return");
        }
    }
}

```

```

// 3. TAPES IN STOCK

if (Choice==3)
{
    System.out.println();
    for (int i = 0; i < 20; i++)
    {
        System.out.println("Movie " + i + ": " + Videos[i] + " copies");
    }
}

// 4. CUSTOMER INFORMATION

if (Choice==4)
{
    System.out.println();
    System.out.println("Please enter a Customer Number between 0 and 49");
    CheckCustomer = BasicIo.readInteger();
    System.out.println();

    if ((CheckCustomer>-1) && (CheckCustomer<50)) // check range
    {
        if (Customers[CheckCustomer] > -1) // if customer has hired video
        {
            System.out.println("Customer " + CheckCustomer + " currently
                has Tape " + Customers[CheckCustomer]);
        }
        else // i.e. Customer has no tape
        {
            System.out.println("Customer " + CheckCustomer + " currently
                has No Tape");
        }
    }
}

while ((Choice<5) && (Choice>0)); // while input is in range specified by menu

/* end of code */

}
}

```

## Exercise 2: Sample Output

```
M:\cafe\Projects Semester 2\Lab12, Exercise 2>java VideoShop
Symantec Java! JustInTime Compiler Version 210.050 for JDK 1.1
Copyright (C) 1996-97 Symantec Corporation
```

```
MAIN MENU - PLEASE ENTER YOUR CHOICE
-----
```

1. Video Hire
2. Video Return
- 3: Tapes in Stock
4. Customer Information

5. Exit

3

```
Movie 0: 4 copies
Movie 1: 4 copies
Movie 2: 4 copies
Movie 3: 4 copies
Movie 4: 4 copies
Movie 5: 4 copies
Movie 6: 4 copies
Movie 7: 4 copies
Movie 8: 4 copies
Movie 9: 4 copies
Movie 10: 6 copies
Movie 11: 6 copies
Movie 12: 6 copies
Movie 13: 6 copies
Movie 14: 6 copies
Movie 15: 6 copies
Movie 16: 6 copies
Movie 17: 6 copies
Movie 18: 6 copies
Movie 19: 6 copies
```

```
MAIN MENU - PLEASE ENTER YOUR CHOICE
-----
```

1. Video Hire
2. Video Return
- 3: Tapes in Stock
4. Customer Information

5. Exit

1

```
Which video do you want to hire?
Enter a number between 0 and 19
```

3

```
Which customer wants to hire this video?
```

```
Enter a number between 0 and 49
```

44

```
MAIN MENU - PLEASE ENTER YOUR CHOICE
-----
```

1. Video Hire
2. Video Return
- 3: Tapes in Stock
4. Customer Information

5. Exit

4

```
Please enter a Customer Number between 0 and 49
```

23

```
Customer 23 currently has No Tape
```

MAIN MENU - PLEASE ENTER YOUR CHOICE

-----

1. Video Hire
2. Video Return
- 3: Tapes in Stock
4. Customer Information

5. Exit

4

Please enter a Customer Number between 0 and 49

44

Customer 44 currently has Tape 3

MAIN MENU - PLEASE ENTER YOUR CHOICE

-----

1. Video Hire
2. Video Return
- 3: Tapes in Stock
4. Customer Information

5. Exit

2

Which Customer is Returning a video?

44

MAIN MENU - PLEASE ENTER YOUR CHOICE

-----

1. Video Hire
2. Video Return
- 3: Tapes in Stock
4. Customer Information

5. Exit

5

## “Extra”: Exercise 1: Program Code

```
/**
 *
 * Lab 12, Exercise 1
 *
 * Written by Gareth Evans
 *
 * 25/Feb/99
 */

import java.util.* ;
import java.bangor.*;

public class ExamResults
{
    /**
     *
     * main
     *
     */

    public static void main(String arg[]) throws Exception
    {

        // Enter the Marks

        int[] examMarks;
        examMarks = new int[12];

        for (int i = 0; i < 12; i++)
        {
            do
            {
                System.out.println("Please enter Exam Mark for exam number " + (i+1));
                examMarks[i] = BasicIo.readInteger();
            }
            while (examMarks[i] < 0  &&  examMarks[i] > 101);
        }

        // Display the Highest Mark

        int Highest;
        Highest = 1;

        for (int k = 1; k < 12; k++)
        {
            if (examMarks[k] > examMarks[(Highest-1)]) // if next in list is higher
            { Highest = k+1; } // than previous high
        }

        System.out.println();
        System.out.println("The Highest mark was mark " + Highest + ", this was " +
            examMarks[Highest-1]);

        System.out.println();
    }
}
```

```

// Display the lowest mark

int Lowest;
Lowest = 1;

for (int l = 1; l < 12; l++)
{
    if (examMarks[l] < examMarks[(Lowest-1)]) // if next in list is lower
        { Lowest = l+1; } // than previous low
}

System.out.println();
System.out.println("The Lowest mark was mark " + Lowest + ", this was " +
examMarks[Lowest-1]);

// Display the average mark

int Sum;
Sum = 0;

for (int m = 0; m < 12; m++)
{
    Sum = Sum + examMarks[m]; // add all the marks up
}

Sum = Sum/12; // divide by the number of pupils

System.out.println();
System.out.println("The average mark was " + Sum);
System.out.println();

// Display how many marks were above 60

int Count;
Count = 0;

for (int n = 0; n < 12; n++)
{
    if (examMarks[n]>60) // if mark is above 60 add 1 to our count
        { Count++; }
}

System.out.println();
System.out.println("There were " + Count + " Mark(s) over 60");

/* end of code */

}
}

```

## “Extra”: Exercise 3: Program Code

```
/**
 *
 * Lab 12, Exercise 3
 * Written by Gareth Evans
 *
 * 02/Mar/99
 */

import java.util.* ;
import java.bangor.*;

public class VideoShop
{
    /**
     * main
     */

    public static void main(String arg[]) throws Exception
    {
        // Define the Variables

        int[] Videos;
        Videos = new int[20];

        String[] VideoNames;
        VideoNames = new String[20];
        for (int tempcount1 = 0; tempcount1 < 20; tempcount1++)
        {
            VideoNames[tempcount1] = new String(); // initialise Strings
        }

        int Choice;
        int HireVideo;
        String HireCustomer = new String();
        int ReturnVideo;
        String ReturnCustomer = new String();
        int CheckVideo;
        String CheckCustomer = new String();

        int VideoHireTemp;
        int VideoReturnTemp;
        int InfoTemp;

        // Note: The Videos variable will store the number of videos in stock
        //       The VideoNames array will store the video names

        int[] Customers;
        Customers = new int[50];

        String[] CustomerNames;
        CustomerNames = new String[50];
        for (int tempcount2 = 0; tempcount2 < CustomerNames.length; tempcount2++)
        {
            CustomerNames[tempcount2] = new String(); // initialise Strings
        }

        // Note: The Customers Array will store which video the customer
        //       Currently has ( the value -1 = no video)
        //       The CustomerNames Array will store the Customers' names.
    }
}
```

```

// Set the Variable Values

for (int i = 0; i < 10; i++)
{ Videos[i] = 4; } // We have 4 copies of Videos 0 to 9

for (int j = 10; j < 20; j++)
{ Videos[j] = 6; } // We have 6 copies of Videos 10 to 19

for (int k = 0; k < 50; k++)
{ Customers[k] = -1; } // No videos are hired out at the start

// Set the video names

VideoNames[0] = "Braveheart";
VideoNames[1] = "Apollo 13";
VideoNames[2] = "Aliens 2";
VideoNames[3] = "Predator";
VideoNames[4] = "Sleepless in Seattle";
VideoNames[5] = "Trainspotting";
VideoNames[6] = "The Jungle Book";
VideoNames[7] = "True Lies";
VideoNames[8] = "Tomorrow Never Dies";
VideoNames[9] = "Speed";
VideoNames[10] = "G.I. Jane";
VideoNames[11] = "Ants";
VideoNames[12] = "Lock, Stock & 2 Smoking Barrels";
VideoNames[13] = "Titanic";
VideoNames[14] = "Independence Day";
VideoNames[15] = "Men In Black";
VideoNames[16] = "X Files The Movie";
VideoNames[17] = "A Bug's Life";
VideoNames[18] = "Saving Private Ryan" ;
VideoNames[19] = "RoboCop";

// Set the Customer names; Note - only 10 included; rest could
// be entered via another option in the menu?

CustomerNames[0] = "Gareth Evans";
CustomerNames[1] = "Jim Smith";
CustomerNames[2] = "Alex Ferguson";
CustomerNames[3] = "Kenny Dalglish";
CustomerNames[4] = "Glenn Hoddle";
CustomerNames[5] = "Bobby Gould";
CustomerNames[6] = "Kevin Keegan";
CustomerNames[7] = "Roy Evans";
CustomerNames[8] = "Walter Smith";
CustomerNames[9] = "Ruud Gullit";

// MENU SYSTEM

do
{
    System.out.println();
    System.out.println("MAIN MENU - PLEASE ENTER YOUR CHOICE");
    System.out.println("-----");
    System.out.println();
    System.out.println("1. Video Hire - (Choose option 3 if unsure which video
                        you want)");
    System.out.println("2. Video Return");
    System.out.println("3: Tapes in Stock");
    System.out.println("4. Customer Information");
    System.out.println();
    System.out.println("5. Exit");
    System.out.println();
    Choice = BasicIo.readInteger();

    // 1. VIDEO HIRE

    if (Choice==1)
    {
        System.out.println("Which video do you want to hire?");
        System.out.println("Enter a number between 0 and 19");
        HireVideo = BasicIo.readInteger();
    }
}

```

```

if (Videos[HireVideo]>0)
{
    System.out.println();
    System.out.println("Which customer wants to hire this video?");
    System.out.println("Enter the Customer's Full Name");
    HireCustomer = BasicIo.readString();

    VideoHireTemp = -1;

    for (int i = 0; i <= 49; i++) // search for customer
    {
        if ( CustomerNames[i].equals(HireCustomer) )
        { VideoHireTemp = i; }
    }

    if (VideoHireTemp == -1)
    { System.out.println("Sorry. There is no customer with that name.");}

    if (VideoHireTemp > -1) // i.e. customer exists
    {
        if ((Customers[VideoHireTemp]) == (-1)) // ie. customer has no video
        {
            Customers[VideoHireTemp] = HireVideo; // give customer video
            Videos[HireVideo] = (Videos[HireVideo] - 1); // decrease stock
        }
        else // i.e. customer already has a video
        {
            System.out.println();
            System.out.println("The Customer selected has already hired a
                                video");
            System.out.println("Please return this video before attempting to
                                hire another video");
        }
    }
}
else // i.e. no copies of video in stock
{
    System.out.println();
    System.out.println("There are no copies of this video in stock.
                        Sorry");
}
}

// 2. VIDEO RETURN

if (Choice==2)
{
    System.out.println();
    System.out.println("Which Customer is Returning a video?");
    System.out.println("Please type in the Full Name");
    ReturnCustomer = BasicIo.readString();

    VideoReturnTemp = -1;

    for (int i = 0; i <= 49; i++) // as before - check customer exists
    {
        if ( CustomerNames[i].equals(ReturnCustomer) )
        { VideoReturnTemp = i; }
    }

    if (VideoReturnTemp == -1)
    { System.out.println("Sorry. There is no customer with that name."); }

    if (VideoReturnTemp > -1) // if customer exists
    {
        if (Customers[VideoReturnTemp]>-1) // if customer has a video
        {
            ReturnVideo = Customers[VideoReturnTemp];
            Videos[ReturnVideo] = (Videos[ReturnVideo] + 1); // increase stock
            Customers[VideoReturnTemp] = -1; // flag customer as having
            // no video
        }
    }
}

```

```

        else
        {
            System.out.println();
            System.out.println("The selected customer has no videos to
                                return");
        }
    }
}

// 3. TAPES IN STOCK

if (Choice==3)
{
    System.out.println();
    for (int i = 0; i < 20; i++)
    {
        System.out.println(i + ": " + VideoNames[i] + ": " + Videos[i] + "
                            copies");
    }
}

// 4. CUSTOMER INFORMATION

if (Choice==4)
{
    System.out.println();
    System.out.println("Please enter a Customer Name (Full Name)");
    CheckCustomer = BasicIo.readString();
    System.out.println();

    InfoTemp = -1;

    for (int i = 0; i <= 49; i++) // check customer exists
    {
        if ( CustomerNames[i].equals(CheckCustomer) )
        { InfoTemp = i; }
    }

    if (InfoTemp == -1)
    { System.out.println("Sorry. There is no customer with that name."); }

    if ((InfoTemp>-1) && (InfoTemp<50)) // if customer # in range allowed
    {
        if (Customers[InfoTemp] > -1)
        {
            System.out.println(CheckCustomer + " currently has " +
                                VideoNames[Customers[InfoTemp]]);
        }
        else
        {
            System.out.println(CheckCustomer + " currently has No Tape");
        }
    }
}

}
while ((Choice<5) && (Choice>0));

/* end of code */

}

```

**Assignment:** “Hand in the *code* you have produced for Exercise 5 of laboratory session 11 together with a printout of the output of the program”.

## Exercise 5: Program Code

```
/*
 *
 * Gareth Evans
 *
 * 18th February 1999 - 25th February 1999
 *
 * Exercise 5
 *
 * Simulation of a 5 race season
 * And assigns "Sponsorship" bonuses
 */

/* "A program that presents some F1 driver information.
   I was inspired by last year's disappointing result" */

import java.util.* ;
import java.bangor.*;

public class GrandPrix98
{

    /**
     *
     * main
     */

    public static void main(String arg[]) throws Exception
    {

        // declare variables

        Fldriver mika = new Fldriver();
        Fldriver michael = new Fldriver();

        int WCBonus;
        WCBonus = 500000;

        int racenumber;
        int input;
        int mikatemp;
        int michaeltemp;
        int mikasponsor=0;
        int michaelsponsor=0;
        int winstemp;
        int championship=length=5;

        // set driver data

        mika.setPoints(0);
        michael.setPoints(0);
        mika.setName("Mike Hakkinen");
        michael.setName("Michael Schemata");
        mika.setWCB(0);
        michael.setWCB(0);

        // simulate the 5 race season

        racenumber=1;
        System.out.println();
    }
}
```

```

for (int count=0; count<championshiplength; count++) // repeat 5 times
{
    System.out.println("This is race number " + racenumber);
    System.out.println();

    mikatemp = mika.getPoints();
    michaeltemp = michael.getPoints();
    BasicIo.prompt("What was Michael Schumacher's Position in the race? ");
    input = BasicIo.readInteger();

    switch (input) // give Michael Schemata points, bonuses and sponsor money
    {
        case 1: michael.setPoints(michaeltemp + Fldriver.firstPlacePoints);
                winstemp = michael.getWins();
                winstemp++;
                michael.setWins(winstemp);
                michaelsponsor=michaelsponsor+10000;
                break;
        case 2: michael.setPoints(michaeltemp + Fldriver.secondPlacePoints);
                michaelsponsor=michaelsponsor+10000;
                break;
        case 3: michael.setPoints(michaeltemp + Fldriver.thirdPlacePoints);
                michaelsponsor=michaelsponsor+10000;
                break;
        case 4: michael.setPoints(michaeltemp + Fldriver.fourthPlacePoints);
                break;
        case 5: michael.setPoints(michaeltemp + Fldriver.fifthPlacePoints);
                break;
        case 6: michael.setPoints(michaeltemp + Fldriver.sixthPlacePoints);
                break;
        default: break;
    } ;

    System.out.println();

    BasicIo.prompt("What was Mike Hakkinen's Position in the race? ");
    input = BasicIo.readInteger();

    switch (input) // give Mike Hakkinen points, bonuses and sponsor money
    {
        case 1: mika.setPoints(mikatemp + Fldriver.firstPlacePoints);
                winstemp = mika.getWins();
                winstemp++;
                mika.setWins(winstemp);
                mikasponsor=mikasponsor+10000;
                break;
        case 2: mika.setPoints(mikatemp + Fldriver.secondPlacePoints);
                mikasponsor=mikasponsor+10000;
                break;
        case 3: mika.setPoints(mikatemp + Fldriver.thirdPlacePoints);
                mikasponsor=mikasponsor+10000;
                break;
        case 4: mika.setPoints(mikatemp + Fldriver.fourthPlacePoints);
                break;
        case 5: mika.setPoints(mikatemp + Fldriver.fifthPlacePoints);
                break;
        case 6: mika.setPoints(mikatemp + Fldriver.sixthPlacePoints);
                break;
        default: break;
    } ;

    // Show the situation

    System.out.println();
    System.out.println("Standings - Race " + racenumber);
    System.out.println();
    System.out.print(mika.getName());
    System.out.print(" - ");
    System.out.println(mika.getPoints() + " PTAs");
    System.out.print(michael.getName());
    System.out.print(" - ");
    System.out.println(michael.getPoints() + " PTAs");
    System.out.println();
}

```

```

// Check to see who is winning the championship

mikatemp=mika.getPoints();
michaeltemp=michael.getPoints();

if (mikatemp>michaeltemp)
{
    System.out.println("Mike Hakkinen is winning the championship");
}

if (michaeltemp>mikatemp)
{
    System.out.println("Michael Schemata is winning the championship");
}

if (michaeltemp == mikatemp) // (IF POINTS ARE EQUAL, CHECK WIN COLUMN)
{
    michaeltemp = michael.getWins();
    mikatemp = mika.getWins();

    if (mikatemp>michaeltemp)
    {
        System.out.println("Mike has won more races than Michael, ");
        System.out.println("Therefore Mike in leading the championship");
    }

    if (michaeltemp>mikatemp)
    {
        System.out.println("Michael has won more races than Mike, ");
        System.out.println("Therefore Michael in leading the championship");
    }

    if (mikatem==michaeltemp)
    {
        System.out.println("It is a draw at this stage!");
    }
}

racenumber++;
System.out.println();
System.out.println();
}

// Now out of the loop, all races finished

System.out.println();
System.out.println("Standings - After the end of our championship ");
System.out.println();
System.out.print(mika.getName());
System.out.print(" - ");
System.out.println(mika.getPoints() + " PTAs");
System.out.print(michael.getName());
System.out.print(" - ");
System.out.println(michael.getPoints() + " PTAs");

// Display who has won the World Championship

if (mika.getPoints()>michael.getPoints())
{
    mika.setWCB(1);
    System.out.println("Mike Hakkinen wins the championship");
}

if (michael.getPoints()>mika.getPoints())
{
    michael.setWCB(1);
    System.out.println("Michael Schemata wins the championship");
}

if (michael.getPoints() == mika.getPoints())
{
    if (mika.getWins()>michael.getWins())
    {
        mika.setWCB(1);
        System.out.println("Mike has won more races than Michael, ");
        System.out.println("Therefore Mike wins the championship");
    }
}

```

```

        if (michael.getWins()>mika.getWins())
        {
            michael.setWCB(1);
            System.out.println("Michael has won more races than Mike, ");
            System.out.println("Therefore Michael wins the championship");
        }

        if (mika.getWins()==michael.getWins())
        {
            System.out.println("It is a draw!");
        }
    }

    // Display World Championship Bonus

    System.out.println();
    System.out.println("Bonuses:");
    System.out.println();

    if ((mika.getWCB()==1)
        System.out.println("Mike Hakkinen Bonus = "+WCBonus+" Pounds");
    else
        System.out.println("Mike Hakkinen Bonus = 0");

    if ((michael.getWCB()==1)
        System.out.println("Michael Schemata Bonus = "+WCBonus+" Pounds");
    else
        System.out.println("Michael Schemata Bonus = 0");

    // Set and Show Sponsorship Bonuses

    mikatemp = mika.getPoints();
    michaeltemp = michael.getPoints();

    if ((mika.getWCB()==1) // if mika is the world champion
        {
            if (mikatemp>69) // if mika has 70 points or more
            {
                if ((mikatemp-michaeltemp)>5) // if the points difference is more than 5
                {
                    int temp;
                    temp = mikatemp-michaeltemp;
                    temp = temp*5000;
                    mikasponsor=mikasponsor+temp;
                }
            }
        }

    if ((michael.getWCB()==1)
        {
            if (michaeltemp>69)
            {
                if ((michaeltemp-mikatemp)>5)
                {
                    int temp;
                    temp = michaeltemp-mikatemp;
                    temp = temp*5000;
                    michaelsponsor=michaelsponsor+temp;
                }
            }
        }
    }

    System.out.println();
    System.out.println("Sponsorship Bonuses:");
    System.out.println();
    System.out.println("Michael Schemata: " + michaelsponsor + " pounds");
    System.out.println("Mike Hakkinen: " + mikasponsor + " pounds");

    /* end of code */
}

```

## Exercise 5: Formula 1 Class: Source Code

```
package java.bangor;

/* definition of a simple class for F1 drivers */

public class F1driver
{
private String name;
private int points;
private int wins;
private int champBonus;

private static int raceNumber;

public static int firstPlacePoints=10;
public static int secondPlacePoints=6;
public static int thirdPlacePoints=4;
public static int fourthPlacePoints=3;
public static int fifthPlacePoints=2;
public static int sixthPlacePoints=1;
public static int otherPlacePoints=0;

public String getName()
{
return name;
}

public void setName(String aName)
{
name = aName;
}

public int getPoints()
{
return points;
}

public void setPoints(int aPoint)
{
points = aPoint;
}

public int getWCB()
{
return champBonus;
}

public void setWCB(int aBonus)
{
champBonus = aBonus;
}

public int getWins()
{
return wins;
}

public void setWins(int aWin)
{
wins = aWin;
}
}
```

## Exercise 5: Sample Output

```
M:\cafe\Projects Semester 2\Lab11, Exercise 5>java GrandPrix98
Symantec Java! JustInTime Compiler Version 210.050 for JDK 1.1
Copyright (C) 1996-97 Symantec Corporation
```

```
This is race number 1
```

```
What was Michael Schumacher's Position in the race? 2
```

```
What was Mike Hakkinen's Position in the race? 4
```

```
Standings - Race 1
```

```
Mike Hakkinen - 3 PTAs
```

```
Michael Schemata - 6 PTAs
```

```
Mike Hakkinen is winning the championship
```

```
This is race number 2
```

```
What was Michael Schumacher's Position in the race? 1
```

```
What was Mike Hakkinen's Position in the race? 3
```

```
Standings - Race 2
```

```
Mike Hakkinen - 7 PTAs
```

```
Michael Schemata - 16 PTAs
```

```
Mike Hakkinen is winning the championship
```

```
This is race number 3
```

```
What was Michael Schumacher's Position in the race? 2
```

```
What was Mike Hakkinen's Position in the race? 3
```

```
Standings - Race 3
```

```
Mike Hakkinen - 11 PTAs
```

```
Michael Schemata - 22 PTAs
```

```
Mike Hakkinen is winning the championship
```

This is race number 4

What was Michael Schumacher's Position in the race? 3

What was Mike Hakkinen's Position in the race? 4

Standings - Race 4

Mike Hakkinen - 14 PTAs

Michael Schemata - 26 PTAs

Mike Hakkinen is winning the championship

This is race number 5

What was Michael Schumacher's Position in the race? 5

What was Mike Hakkinen's Position in the race? 6

Standings - Race 5

Mike Hakkinen - 15 PTAs

Michael Schemata - 28 PTAs

Mike Hakkinen is winning the championship

Standings - After the end of our championship

Mike Hakkinen - 15 PTAs

Michael Schemata - 28 PTAs

Mike Hakkinen wins the championship

Bonuses:

Mike Hakkinen Bonus = 0

Michael Schemata Bonus = 500000 Pounds