**Question No:1**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class TeamDetails

{

private String team;

private String city;

TeamDetails(String t,String c)

{

this.team=t;

this.city=c;

}

void display()

{

System.out.println("Team Details -");

System.out.println("Team: "+team);

System.out.println("City: "+city);

}

}

public Sample Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println

String team=s.nextLine();

String city=s.nextLine();

TeamDetails teamdetails=new TeamDetails(team,city);

teamdetails.display();

}

}

Sample Input and Output:

Enter team name: Platypups

Enter city represented: Krematore

Team Details -

Team: Platypups

City: Krematore

**Question No:2**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class PlayerDetails

{

private String player;

private String playerposition;

PlayerDetails(String p,String pp)

{

this.player=p;

this.playerposition=pp;

}

void display()

{

System.out.println("Player Details -");

System.out.println("player: "+player);

System.out.println("playerposition: "+playerposition);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter player name: ");

String player=s.nextLine();

System.out.println("Enter player position: ");

String playerposition=s.nextLine();

PlayerDetails playerdetail=new PlayerDetails(player,playerposition);

playerdetail.display();

}

}

Sample Input and Output:

Enter player name: Phiniboms

Enter player position: Speedster

Player Details -

Player: Phiniboms

Position: Speedster

**Question No:3**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class MatchDetails

{

private int match;

private int win;

private int loss;

private float avg\_lap\_speed;

private float max\_lap\_speed;

private float avg\_throw\_speed;

private float max\_throw\_speed;

MatchDetails(int match,int win,int loss,float avg\_lap\_speed,float max\_lap\_speed,float avg\_throw\_speed,float max\_throw\_speed)

{

this.match=match;

this.win=win;

this.loss=loss;

this.avg\_lap\_speed=avg\_lap\_speed;

this.max\_lap\_speed=max\_lap\_speed;

this.avg\_throw\_speed=avg\_throw\_speed;

this.max\_throw\_speed=max\_throw\_speed;

}

void display()

{

System.out.println("Player Record -");

System.out.println("Matches Played: "+match);

System.out.println("Win/Loss Ratio: "+(win/2)+":"+(loss/2));

System.out.println("Avg/Max Lap Speed: "+avg\_lap\_speed+" | "+max\_lap\_speed+"min");

System.out.println("Avg/Max Throw Distance: "+avg\_throw\_speed+" | "+max\_throw\_speed+"m");

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter matches: ");

int match=s.nextInt();

System.out.println("Enter Wins: ");

int win=s.nextInt();

System.out.println("Enter losses: ");

int loss=s.nextInt();

System.out.println("Enter avg lap speed: ");

float avg\_lap\_speed=s.nextFloat();

System.out.println("Enter max lap speed: ");

float max\_lap\_speed=s.nextFloat();

System.out.println("Enter avg throw distance: ");

float avg\_throw\_speed=s.nextFloat();

System.out.println("Enter max throw distance: ");

float max\_throw\_speed=s.nextFloat();

MatchDetails matchdetails=new MatchDetails(match,win,loss,avg\_lap\_speed,max\_lap\_speed,avg\_throw\_speed,max\_throw\_speed);

matchdetails.display();

}

}

Sample Input and Output:

Enter matches: 58

Enter wins: 42

Enter losses: 16

Enter avg lap speed: 12.50

Enter max lap speed: 8.79

Enter avg throw distance: 50.15

Enter max throw distance: 62.44

Player Record -

Matches Played: 58

Win/Loss Ratio: 21:8

Avg/Max Lap Speed: 12.50 | 8.79 min

Avg/Max Throw Distance: 50.15 | 62.44 m

**Question No:4**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class Team

{

private String name;

private String city;

public void setter\_name(String name)

{

this.name=name;

}

public String getter\_name()

{

return name;

}

public void setter\_city(String city)

{

this.city=city;

}

public String getter\_city()

{

return city;

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter team name: ");

String name1=s.nextLine();

System.out.println("Enter city represented: ");

String city1=s.nextLine();

String name="",city="";

Team team=new Team();

System.out.println("VERIFY AND UPDATE DETAILS");

System.out.println("1. Update Team Name");

System.out.println("2. Update City Represented");

System.out.println("3. Save and Exit");

System.out.println("Enter choice: ");

int choice=s.nextInt();

switch(choice)

{

case 1:

System.out.println("Enter team name: ");

s.nextLine();

name=s.nextLine();

team.setter\_name(name);

System.out.println("Team: "+team.getter\_name());

break;

case 2:

System.out.println("Enter city represented");

s.nextLine();

city=s.nextLine();

team.setter\_city(name);

System.out.println("City: "+team.getter\_city());

break;

case 3:

System.out.println("Thank you!");

System.exit(0);

}

}

}

Sample Input and Output:

Enter team name: Platypups

Enter city represented: Krematore

VERIFY AND UPDATE DETAILS

1. Update Team Name

2. Update City Represented

3. Save and Exit

Enter choice: 1

Enter team name: Pulipups

Team: Pulipups

City: Krematore

VERIFY AND UPDATE DETAILS

1. Update Team Name

2. Update City Represented

3. Save and Exit

Enter choice: 3

Thank you!

**Question No:5**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

Node head;

static class Node

{

int throwmade;

String fastest\_player;

float time;

Node next;

Node(int throwmade,String fastest\_player,float time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.time=time;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

Node temp=lap.head;

int c=1;

System.out.println("Lap Details");

while(temp!=null)

{

System.out.println("Lap "+c);

System.out.println("Throws: "+temp.throwmade);

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.time+"s");

c++;

temp=temp.next;

}

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of Laps: ");

LapDetails lap=new LapDetails();

int n=s.nextInt();

int throwmade;

String fastest\_player;

float time;

for(int i=1;i<=n;i++)

{

System.out.println("Enter the details of lap "+i);

throwmade=s.nextInt();

s.nextLine();

fastest\_player=s.nextLine();

time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,time);

}

lap.display(lap);

}

}

Sample Input and Output:

Enter the number of laps: 3

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Lap Details

Lap: 1

Throws: 15

Fastest Player: Hekinsly

Fastest Time: 9.75 s

Lap: 2

Throws: 10

Fastest Player: Probian

Fastest Time: 10.15 s

Lap: 3

Throws: 11

Fastest Player: Satterfly

Fastest Time: 9.13 s

**Question No:6**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class Player

{

private String playername;

private String playerposition;

public void setPlayerName(String name)

{

this.playername=name;

}

public void setPlayerPosition(String name)

{

this.playerposition=name;

}

public String getPlayerName()

{

return playername;

}

public String getPlayerPosition()

{

return playerposition;

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

Player p=new Player();

System.out.println("Enter player name: ");

String name=s.nextLine();

System.out.println("Enter player position: ");

String position=s.nextLine();

p.setPlayerName(name);

p.setPlayerPosition(position);

System.out.println("class Player [ "+p.getPlayerName()+" --- "+p.getPlayerPosition()+" ]");

}

}

Sample Input and Output:

Enter player name: Phiniboms

Enter player position: Speedster

Player Details -

class Player [ Phiniboms --- Speedster ]

**Question No:7**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of laps:");

int n=s.nextInt();

int throwmade;

String fastest\_player;

float fastest\_time;

LapDetails lap=new LapDetails();

for(int i=1;i<=n;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time);

}

lap.display(lap);

}

}

Sample Input and Output:

Enter the number of laps: 3

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Lap Details

LAP THROWS FASTEST PLAYER TIME

1 15 Hekinsly 9.75s

2 10 Probian 10.15s

3 11 Satterfly 9.13s

**Question No:8**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

class LapManager

{

int lap\_no;

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

this.lap\_no=lap\_no;

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time,lap\_no);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

//System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

public void fastest(LapDetails lap,int i)

{

LapManager temp=lap.head;

int j=1;

while(temp!=null)

{

if(j!=i)

{

temp=temp.next;

j++;

}else

{

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.fastest\_time);

break;

}

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of laps:");

int n=s.nextInt();

int throwmade;

String fastest\_player;

float fastest\_time;

LapDetails lap=new LapDetails();

for(int i=1;i<=n;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time,i);

}

System.out.println("LAP MANAGER");

System.out.println("1. Lap Details");

System.out.println("2. Fastest in Lap");

System.out.println("3. Maximum Throws");

System.out.println("4. Exit");

System.out.println("Enter choice: ");

int i;

int choice=s.nextInt();

switch(choice)

{

case 1:

//System.out.println("LAP THROWS FASTEST PLAYER TIME");

lap.display(lap);

break;

case 2:

System.out.println("Enter lap no: ");

i=s.nextInt();

if(i>n)

{

System.out.println("Invalid lap number");

System.exit(0);

}

lap.fastest(lap,i);

break;

case 3:

lap.maximum(lap);

break;

case 4:

System.out.println("Thank you");

System.exit(0);

}

}

}

Sample Input and Output:

Enter the number of laps: 3

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 1

LAP THROWS FASTEST PLAYER TIME

1 15 Hekinsly 9.75s

2 10 Probian 10.15s

3 11 Satterfly 9.13s

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 2

Enter lap no: 2

Fastest Player: Probian

Fastest Time: 10.15s

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 3

15 throws were made in lap 1

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 4

Thank you!

**Question No:9**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

LapManager head1;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager connection;

int m;

String name;

LapManager next1;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

LapManger(int m,String name)

{

this.m=m;

this.name=name;

next1=null;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time,lap\_no);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void setPlayerName(String name)

{

this.playername=name;

}

public void setPlayerPosition(String name)

{

this.playerposition=name;

}

public String getPlayerName()

{

return playername;

}

public String getPlayerPosition()

{

return playerposition;

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

public void insert1(LapDetails lap,int m,int name)

{

LapManager newnode=new LapManager()

public void fastest(LapDetails lap,int i)

{

LapManager temp=lap.head;

int j=1;

while(temp!=null)

{

if(j!=i)

{

temp=temp.next;

j++;

}else

{

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.fastest\_time);

break;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

}

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int throwmade,i,j,n;

String name;

String fastest\_player;

float fastest\_time;

LapDetails lap=new LapDetails();

LapDetails lap1=new LapDetails();

System.out.println("Enter the number of race:");

int n=s.nextInt();

for(i=1;i<=n;i++)

{

System.out.println("Enter the details of race"+i)

m=s.nextInt();

name=s.next();

for( i=1;i<=m;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time);

switch(choice)

{

case 1:

//System.out.println("LAP THROWS FASTEST PLAYER TIME");

lap.display(lap);

break;

case 2:

System.out.println("Enter lap no: ");

i=s.nextInt();

if(i>n)

{

System.out.println("Invalid lap number");

System.exit(0);

}

lap.fastest(lap,i);

break;

case 3:

lap.maximum(lap);

break;

case 4:

System.out.println("Thank you");

System.exit(0);

}

}

lap1.insert1(lap,m,name);

}

lap.display(lap);

}

}

Sample Input and Output:

Enter the number of races: 5

Enter details of race 1:

3, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of race 2:

5, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of lap 4:

11, Satterfly, 9.13

Enter details of lap 5:

11, Satterfly, 9.13

Enter details of race 3

3, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of race 4:

3, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of race 5:

7, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of lap 4:

11, Satterfly, 9.13

Enter details of lap 5:

11, Satterfly, 9.13

Enter details of lap 6:

11, Satterfly, 9.13

Enter details of lap 7:

11, Satterfly, 9.13

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 1

RACE NO LAPS WINNER (PLAYER)

1 3 Hekinsly

2 5 Probian

3 3 Hekinsly

4 3 Satterfly

5 7 Satterfly

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 2

Enter lap count: 7

RACE NO LAPS WINNER (PLAYER)

5 7 Satterfly

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 3

Enter player name: Probian

RACE NO LAPS WINNER (PLAYER)

2 5 Probian

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 4

Race no: 5

LAP THROWS FASTEST PLAYER TIME

1 15 Hekinsly 9.75s

2 10 Probian 10.15s

3 11 Satterfly 9.13s

4 10 Probian 10.15s

5 11 Satterfly 9.13s

6 10 Probian 10.15s

7 11 Satterfly 9.13s

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 5Question No:1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class TeamDetails

{

private String team;

private String city;

TeamDetails(String t,String c)

{

this.team=t;

this.city=c;

}

void display()

{

System.out.println("Team Details -");

System.out.println("Team: "+team);

System.out.println("City: "+city);

}

}

public Sample Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println

String team=s.nextLine();

String city=s.nextLine();

TeamDetails teamdetails=new TeamDetails(team,city);

teamdetails.display();

}

}

Sample Input and Output:

Enter team name: Platypups

Enter city represented: Krematore

Team Details -

Team: Platypups

City: Krematore

Question No:2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class PlayerDetails

{

private String player;

private String playerposition;

PlayerDetails(String p,String pp)

{

this.player=p;

this.playerposition=pp;

}

void display()

{

System.out.println("Player Details -");

System.out.println("player: "+player);

System.out.println("playerposition: "+playerposition);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter player name: ");

String player=s.nextLine();

System.out.println("Enter player position: ");

String playerposition=s.nextLine();

PlayerDetails playerdetail=new PlayerDetails(player,playerposition);

playerdetail.display();

}

}

Sample Input and Output:

Enter player name: Phiniboms

Enter player position: Speedster

Player Details -

Player: Phiniboms

Position: Speedster

Question No:3

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class MatchDetails

{

private int match;

private int win;

private int loss;

private float avg\_lap\_speed;

private float max\_lap\_speed;

private float avg\_throw\_speed;

private float max\_throw\_speed;

MatchDetails(int match,int win,int loss,float avg\_lap\_speed,float max\_lap\_speed,float avg\_throw\_speed,float max\_throw\_speed)

{

this.match=match;

this.win=win;

this.loss=loss;

this.avg\_lap\_speed=avg\_lap\_speed;

this.max\_lap\_speed=max\_lap\_speed;

this.avg\_throw\_speed=avg\_throw\_speed;

this.max\_throw\_speed=max\_throw\_speed;

}

void display()

{

System.out.println("Player Record -");

System.out.println("Matches Played: "+match);

System.out.println("Win/Loss Ratio: "+(win/2)+":"+(loss/2));

System.out.println("Avg/Max Lap Speed: "+avg\_lap\_speed+" | "+max\_lap\_speed+"min");

System.out.println("Avg/Max Throw Distance: "+avg\_throw\_speed+" | "+max\_throw\_speed+"m");

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter matches: ");

int match=s.nextInt();

System.out.println("Enter Wins: ");

int win=s.nextInt();

System.out.println("Enter losses: ");

int loss=s.nextInt();

System.out.println("Enter avg lap speed: ");

float avg\_lap\_speed=s.nextFloat();

System.out.println("Enter max lap speed: ");

float max\_lap\_speed=s.nextFloat();

System.out.println("Enter avg throw distance: ");

float avg\_throw\_speed=s.nextFloat();

System.out.println("Enter max throw distance: ");

float max\_throw\_speed=s.nextFloat();

MatchDetails matchdetails=new MatchDetails(match,win,loss,avg\_lap\_speed,max\_lap\_speed,avg\_throw\_speed,max\_throw\_speed);

matchdetails.display();

}

}

Sample Input and Output:

Enter matches: 58

Enter wins: 42

Enter losses: 16

Enter avg lap speed: 12.50

Enter max lap speed: 8.79

Enter avg throw distance: 50.15

Enter max throw distance: 62.44

Player Record -

Matches Played: 58

Win/Loss Ratio: 21:8

Avg/Max Lap Speed: 12.50 | 8.79 min

Avg/Max Throw Distance: 50.15 | 62.44 m

Question No:4

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class Team

{

private String name;

private String city;

public void setter\_name(String name)

{

this.name=name;

}

public String getter\_name()

{

return name;

}

public void setter\_city(String city)

{

this.city=city;

}

public String getter\_city()

{

return city;

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter team name: ");

String name1=s.nextLine();

System.out.println("Enter city represented: ");

String city1=s.nextLine();

String name="",city="";

Team team=new Team();

System.out.println("VERIFY AND UPDATE DETAILS");

System.out.println("1. Update Team Name");

System.out.println("2. Update City Represented");

System.out.println("3. Save and Exit");

System.out.println("Enter choice: ");

int choice=s.nextInt();

switch(choice)

{

case 1:

System.out.println("Enter team name: ");

s.nextLine();

name=s.nextLine();

team.setter\_name(name);

System.out.println("Team: "+team.getter\_name());

break;

case 2:

System.out.println("Enter city represented");

s.nextLine();

city=s.nextLine();

team.setter\_city(name);

System.out.println("City: "+team.getter\_city());

break;

case 3:

System.out.println("Thank you!");

System.exit(0);

}

}

}

Sample Input and Output:

Enter team name: Platypups

Enter city represented: Krematore

VERIFY AND UPDATE DETAILS

1. Update Team Name

2. Update City Represented

3. Save and Exit

Enter choice: 1

Enter team name: Pulipups

Team: Pulipups

City: Krematore

VERIFY AND UPDATE DETAILS

1. Update Team Name

2. Update City Represented

3. Save and Exit

Enter choice: 3

Thank you!

Question No:5

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

Node head;

static class Node

{

int throwmade;

String fastest\_player;

float time;

Node next;

Node(int throwmade,String fastest\_player,float time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.time=time;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

Node temp=lap.head;

int c=1;

System.out.println("Lap Details");

while(temp!=null)

{

System.out.println("Lap "+c);

System.out.println("Throws: "+temp.throwmade);

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.time+"s");

c++;

temp=temp.next;

}

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of Laps: ");

LapDetails lap=new LapDetails();

int n=s.nextInt();

int throwmade;

String fastest\_player;

float time;

for(int i=1;i<=n;i++)

{

System.out.println("Enter the details of lap "+i);

throwmade=s.nextInt();

s.nextLine();

fastest\_player=s.nextLine();

time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,time);

}

lap.display(lap);

}

}

Sample Input and Output:

Enter the number of laps: 3

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Lap Details

Lap: 1

Throws: 15

Fastest Player: Hekinsly

Fastest Time: 9.75 s

Lap: 2

Throws: 10

Fastest Player: Probian

Fastest Time: 10.15 s

Lap: 3

Throws: 11

Fastest Player: Satterfly

Fastest Time: 9.13 s

Question No:6

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class Player

{

private String playername;

private String playerposition;

public void setPlayerName(String name)

{

this.playername=name;

}

public void setPlayerPosition(String name)

{

this.playerposition=name;

}

public String getPlayerName()

{

return playername;

}

public String getPlayerPosition()

{

return playerposition;

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

Player p=new Player();

System.out.println("Enter player name: ");

String name=s.nextLine();

System.out.println("Enter player position: ");

String position=s.nextLine();

p.setPlayerName(name);

p.setPlayerPosition(position);

System.out.println("class Player [ "+p.getPlayerName()+" --- "+p.getPlayerPosition()+" ]");

}

}

Sample Input and Output:

Enter player name: Phiniboms

Enter player position: Speedster

Player Details -

class Player [ Phiniboms --- Speedster ]

Question No:7

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of laps:");

int n=s.nextInt();

int throwmade;

String fastest\_player;

float fastest\_time;

LapDetails lap=new LapDetails();

for(int i=1;i<=n;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time);

}

lap.display(lap);

}

}

Sample Input and Output:

Enter the number of laps: 3

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Lap Details

LAP THROWS FASTEST PLAYER TIME

1 15 Hekinsly 9.75s

2 10 Probian 10.15s

3 11 Satterfly 9.13s

Question No:8

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

class LapManager

{

int lap\_no;

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

this.lap\_no=lap\_no;

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time,lap\_no);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

//System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

public void fastest(LapDetails lap,int i)

{

LapManager temp=lap.head;

int j=1;

while(temp!=null)

{

if(j!=i)

{

temp=temp.next;

j++;

}else

{

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.fastest\_time);

break;

}

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of laps:");

int n=s.nextInt();

int throwmade;

String fastest\_player;

float fastest\_time;

LapDetails lap=new LapDetails();

for(int i=1;i<=n;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time,i);

}

System.out.println("LAP MANAGER");

System.out.println("1. Lap Details");

System.out.println("2. Fastest in Lap");

System.out.println("3. Maximum Throws");

System.out.println("4. Exit");

System.out.println("Enter choice: ");

int i;

int choice=s.nextInt();

switch(choice)

{

case 1:

//System.out.println("LAP THROWS FASTEST PLAYER TIME");

lap.display(lap);

break;

case 2:

System.out.println("Enter lap no: ");

i=s.nextInt();

if(i>n)

{

System.out.println("Invalid lap number");

System.exit(0);

}

lap.fastest(lap,i);

break;

case 3:

lap.maximum(lap);

break;

case 4:

System.out.println("Thank you");

System.exit(0);

}

}

}

Sample Input and Output:

Enter the number of laps: 3

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 1

LAP THROWS FASTEST PLAYER TIME

1 15 Hekinsly 9.75s

2 10 Probian 10.15s

3 11 Satterfly 9.13s

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 2

Enter lap no: 2

Fastest Player: Probian

Fastest Time: 10.15s

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 3

15 throws were made in lap 1

LAP MANAGER

1. Lap Details

2. Fastest in Lap

3. Maximum Throws

4. Exit

Enter choice: 4

Thank you!

Question No:9

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

LapManager head1;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager connection;

int m;

String name;

LapManager next1;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

LapManger(int m,String name)

{

this.m=m;

this.name=name;

next1=null;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time,lap\_no);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void setPlayerName(String name)

{

this.playername=name;

}

public void setPlayerPosition(String name)

{

this.playerposition=name;

}

public String getPlayerName()

{

return playername;

}

public String getPlayerPosition()

{

return playerposition;

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

public void insert1(LapDetails lap,int m,int name)

{

LapManager newnode=new LapManager()

public void fastest(LapDetails lap,int i)

{

LapManager temp=lap.head;

int j=1;

while(temp!=null)

{

if(j!=i)

{

temp=temp.next;

j++;

}else

{

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.fastest\_time);

break;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

}

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int throwmade,i,j,n;

String name;

String fastest\_player;

float fastest\_time;

LapDetails lap=new LapDetails();

LapDetails lap1=new LapDetails();

System.out.println("Enter the number of race:");

int n=s.nextInt();

for(i=1;i<=n;i++)

{

System.out.println("Enter the details of race"+i)

m=s.nextInt();

name=s.next();

for( i=1;i<=m;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time);

switch(choice)

{

case 1:

//System.out.println("LAP THROWS FASTEST PLAYER TIME");

lap.display(lap);

break;

case 2:

System.out.println("Enter lap no: ");

i=s.nextInt();

if(i>n)

{

System.out.println("Invalid lap number");

System.exit(0);

}

lap.fastest(lap,i);

break;

case 3:

lap.maximum(lap);

break;

case 4:

System.out.println("Thank you");

System.exit(0);

}

}

lap1.insert1(lap,m,name);

}

lap.display(lap);

}

}

Sample Input and Output:

Enter the number of races: 5

Enter details of race 1:

3, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of race 2:

5, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of lap 4:

11, Satterfly, 9.13

Enter details of lap 5:

11, Satterfly, 9.13

Enter details of race 3

3, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of race 4:

3, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of race 5:

7, Hekinsly

Enter details of lap 1:

15, Hekinsly, 9.75

Enter details of lap 2:

10, Probian, 10.15

Enter details of lap 3:

11, Satterfly, 9.13

Enter details of lap 4:

11, Satterfly, 9.13

Enter details of lap 5:

11, Satterfly, 9.13

Enter details of lap 6:

11, Satterfly, 9.13

Enter details of lap 7:

11, Satterfly, 9.13

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 1

RACE NO LAPS WINNER (PLAYER)

1 3 Hekinsly

2 5 Probian

3 3 Hekinsly

4 3 Satterfly

5 7 Satterfly

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 2

Enter lap count: 7

RACE NO LAPS WINNER (PLAYER)

5 7 Satterfly

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 3

Enter player name: Probian

RACE NO LAPS WINNER (PLAYER)

2 5 Probian

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 4

Race no: 5

LAP THROWS FASTEST PLAYER TIME

1 15 Hekinsly 9.75s

2 10 Probian 10.15s

3 11 Satterfly 9.13s

4 10 Probian 10.15s

5 11 Satterfly 9.13s

6 10 Probian 10.15s

7 11 Satterfly 9.13s

RACE MANAGER -

1. Display all races

2. Display races with lap count

3. Display races won by (player)

4. Display lap details for race no

5. Exit

Enter choice: 5

Thank you!

Question No:10

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

class LapManager

{

int lap\_no;

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

this.lap\_no=lap\_no;

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

}

class LapDetails

{

LapManager head;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

public void setter\_name(String name)

{

this.name=name;

}

public String getter\_name()

{

return name;

}

public void setter\_city(String city)

{

this.city=city;

}

public String getter\_city()

{

return city;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

Node temp=lap.head;

int c=1;

System.out.println("Lap Details");

while(temp!=null)

{

System.out.println("Lap "+c);

System.out.println("Throws: "+temp.throwmade);

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.time+"s");

c++;

temp=temp.next;

}

}

Node(int throwmade,String fastest\_player,float time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.time=time;

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void fastest(LapDetails lap,int i)

{

LapManager temp=lap.head;

int j=1;

while(temp!=null)

{

if(j!=i)

{

temp=temp.next;

j++;

}else

{

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.fastest\_time);

break;

}

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

}

class MatchDetails

{

private int match;

private int win;

private int loss;

private float avg\_lap\_speed;

private float max\_lap\_speed;

private float avg\_throw\_speed;

private float max\_throw\_speed;

MatchDetails(int match,int win,int loss,float avg\_lap\_speed,float max\_lap\_speed,float avg\_throw\_speed,float max\_throw\_speed)

{

this.match=match;

this.win=win;

this.loss=loss;

this.avg\_lap\_speed=avg\_lap\_speed;

this.max\_lap\_speed=max\_lap\_speed;

this.avg\_throw\_speed=avg\_throw\_speed;

this.max\_throw\_speed=max\_throw\_speed;

}

void display()

{

System.out.println("Player Record -");

System.out.println("Matches Played: "+match);

System.out.println("Win/Loss Ratio: "+(win/2)+":"+(loss/2));

System.out.println("Avg/Max Lap Speed: "+avg\_lap\_speed+" | "+max\_lap\_speed+"min");

System.out.println("Avg/Max Throw Distance: "+avg\_throw\_speed+" | "+max\_throw\_speed+"m");

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

}

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of laps:");

int n=s.nextInt();

int throwmade;

String fastest\_player;

float fastest\_time;

System.out.println("VERIFY AND UPDATE DETAILS");

System.out.println("1. Update Team Name");

System.out.println("2. Update City Represented");

System.out.println("3. Save and Exit");

System.out.println("Enter choice: ");

int choice=s.nextInt();

switch(choice)

{

case 1:

System.out.println("Enter team name: ");

s.nextLine();

name=s.nextLine();

team.setter\_name(name);

System.out.println("Team: "+team.getter\_name());

break;

case 2:

System.out.println("Enter city represented");

s.nextLine();

city=s.nextLine();

team.setter\_city(name);

System.out.println("City: "+team.getter\_city());

break;

case 3:

System.out.println("Thank you!");

System.exit(0);

}

LapDetails lap=new LapDetails();

for(int i=1;i<=n;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time);

}

lap.display(lap);

}

}

Thank you!

**Question No:10**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class LapDetails

{

LapManager head;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

class LapManager

{

int lap\_no;

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time,int lap\_no)

{

this.lap\_no=lap\_no;

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

}

class LapDetails

{

LapManager head;

class LapManager

{

int throwmade;

String fastest\_player;

float fastest\_time;

LapManager next;

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

public void setter\_name(String name)

{

this.name=name;

}

public String getter\_name()

{

return name;

}

public void setter\_city(String city)

{

this.city=city;

}

public String getter\_city()

{

return city;

}

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void display(LapDetails lap)

{

Node temp=lap.head;

int c=1;

System.out.println("Lap Details");

while(temp!=null)

{

System.out.println("Lap "+c);

System.out.println("Throws: "+temp.throwmade);

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.time+"s");

c++;

temp=temp.next;

}

}

Node(int throwmade,String fastest\_player,float time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.time=time;

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float fastest\_time)

{

LapManager newnode=new LapManager(throwmade,fastest\_player,fastest\_time);

LapManager temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

public void fastest(LapDetails lap,int i)

{

LapManager temp=lap.head;

int j=1;

while(temp!=null)

{

if(j!=i)

{

temp=temp.next;

j++;

}else

{

System.out.println("Fastest Player: "+temp.fastest\_player);

System.out.println("Fastest Time: "+temp.fastest\_time);

break;

}

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

public void display(LapDetails lap)

{

LapManager temp=lap.head;

System.out.println("Lap Details");

System.out.println("LAP THROWS FASTEST PLAYER TIME");

int c=1;

while(temp!=null)

{

System.out.println(c+" "+temp.throwmade+" "+temp.fastest\_player+" "+temp.fastest\_time);

c++;

temp=temp.next;

}

}

}

class MatchDetails

{

private int match;

private int win;

private int loss;

private float avg\_lap\_speed;

private float max\_lap\_speed;

private float avg\_throw\_speed;

private float max\_throw\_speed;

MatchDetails(int match,int win,int loss,float avg\_lap\_speed,float max\_lap\_speed,float avg\_throw\_speed,float max\_throw\_speed)

{

this.match=match;

this.win=win;

this.loss=loss;

this.avg\_lap\_speed=avg\_lap\_speed;

this.max\_lap\_speed=max\_lap\_speed;

this.avg\_throw\_speed=avg\_throw\_speed;

this.max\_throw\_speed=max\_throw\_speed;

}

void display()

{

System.out.println("Player Record -");

System.out.println("Matches Played: "+match);

System.out.println("Win/Loss Ratio: "+(win/2)+":"+(loss/2));

System.out.println("Avg/Max Lap Speed: "+avg\_lap\_speed+" | "+max\_lap\_speed+"min");

System.out.println("Avg/Max Throw Distance: "+avg\_throw\_speed+" | "+max\_throw\_speed+"m");

}

public LapDetails insert(LapDetails lap,int throwmade,String fastest\_player,float time)

{

Node newnode=new Node(throwmade,fastest\_player,time);

Node temp=lap.head;

if(lap.head==null)

{

lap.head=newnode;

}else

{

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

}

return lap;

}

}

LapManager(int throwmade,String fastest\_player,float fastest\_time)

{

this.throwmade=throwmade;

this.fastest\_player=fastest\_player;

this.fastest\_time=fastest\_time;

next=null;

}

}

public void maximum(LapDetails lap)

{

LapManager temp=lap.head;

int max=0;

while(temp!=null)

{

if(max<temp.throwmade)

{

max=temp.throwmade;

}

}

System.out.println(temp.throwmade+" throws were made in lap "+temp.lap\_no);

}

}

public class Main

{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the number of laps:");

int n=s.nextInt();

int throwmade;

String fastest\_player;

float fastest\_time;

System.out.println("VERIFY AND UPDATE DETAILS");

System.out.println("1. Update Team Name");

System.out.println("2. Update City Represented");

System.out.println("3. Save and Exit");

System.out.println("Enter choice: ");

int choice=s.nextInt();

switch(choice)

{

case 1:

System.out.println("Enter team name: ");

s.nextLine();

name=s.nextLine();

team.setter\_name(name);

System.out.println("Team: "+team.getter\_name());

break;

case 2:

System.out.println("Enter city represented");

s.nextLine();

city=s.nextLine();

team.setter\_city(name);

System.out.println("City: "+team.getter\_city());

break;

case 3:

System.out.println("Thank you!");

System.exit(0);

}

LapDetails lap=new LapDetails();

for(int i=1;i<=n;i++)

{

System.out.println("Enter details of lap "+i+":");

throwmade=s.nextInt();

fastest\_player=s.next();

fastest\_time=s.nextFloat();

lap.insert(lap,throwmade,fastest\_player,fastest\_time);

}

lap.display(lap);

}

}