



اَوْنِوَرُ سَيِّتِي تَيْكُونُ لُوْكَى مَبَارَا
UNIVERSITI
TEKNOLOGI
MARA

CSC 238 - OBJECT ORIENTED PROGRAMMING

LECTURE NAME:

Madam Zuriati binti Ismail @ Khorl

TITLE:

SPORTS CENTRE SYSTEM

GROUP:

JCS1103F

SECTION:

FINAL REPORT

NAME	STUDENT'S ID
Nurain Nabilah binti Salehuddin	2018258486
Muhammad Aqmar bin Azmi Nor	2018247726

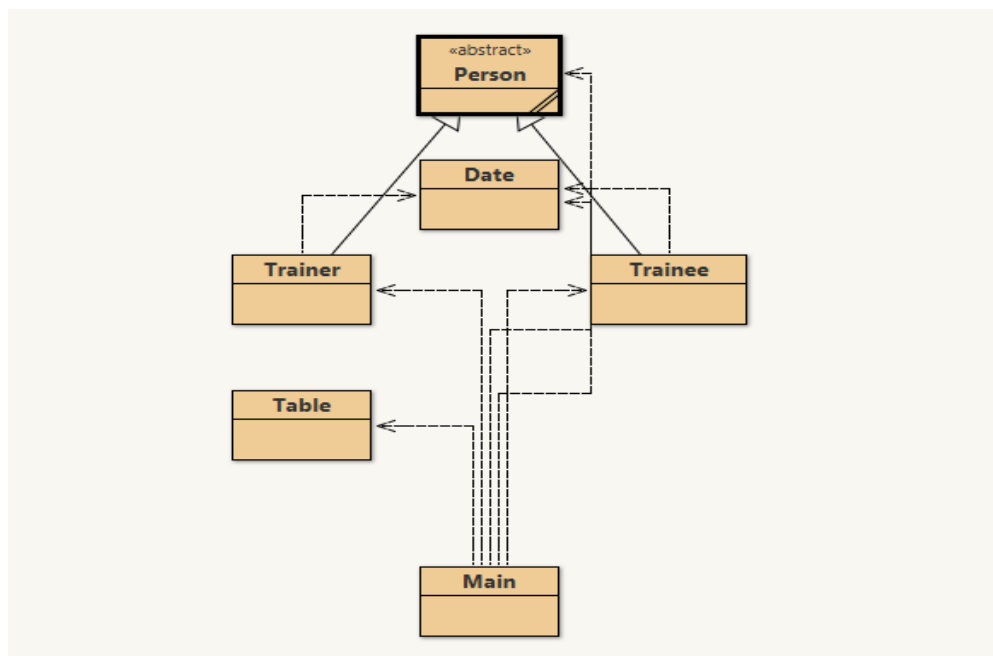
INTRODUCTION

First and foremost, we would like to thank our beloved, Madam Zuriati Bt. Ismail @ Khori for guiding us with this Object Oriented Programming assignment. Hence, this project is called "SDS Centre (Self-Defense Sports Centre)" and it teaches the art of self-defense for example, taekwondo, kendo, silat and boxing. This training centre focuses mainly on teenagers and young adults. The facilities of the sports are provided in the same centre. The aim of this system is to ease the trainees (customers) and trainers (staffs) to register newcomers, check trainers' salary, and display trainees' fee.

OBJECTIVES

1. Stores data of the trainee and trainers.
2. To calculate trainees' fee and trainers' salary.
3. To find minimum fee of the trainees.
4. To find maximum salary of the trainers.
5. To calculate grand total of fee (trainee) and grand total of salary (trainers).
6. To calculate average of grand fee and average of grand salary.
7. To count the number of trainee with fee less than RM200.
8. To count the number of trainers with salary more than RM2100.
9. To sort out the age of all (trainee and trainer) from youngest to oldest
10. To search and display information of the trainees and trainers by using their ID Number.
11. To able to apply I/O file system.
12. To apply usage of constructors, mutators, accessors and processor methods.
13. To apply usage of inheritance and polymorphism in the coding.

CLASS DESIGN



Superclass: Person		
Attributes	-name -ic -phone -email -address	-sport -session -ID -level
Method(s)	- Person() - Person(<i>parameters</i>) - getName() - getIC() - getPhone() - getEmail() - getAddress() - toString()	- getSport() - getLevel() - getSessions() - getuniqueID() - getAge() - setData(<i>parameters</i>) - Elaun()
Subclass: Trainee		
Attributes	- registerDate - member	
Method(s)	- Trainee() - Tainee(<i>parameters</i>) - getRegisterDate() - getMember() - setData (<i>parameters</i>)	- selectionPrice() - calculatePriceAfterDiscount() - Elaun() - displayLevel() - toString()
Subclass: Trainer		
Attributes	- registerDate - member - experience (years) - basic (salary)	
Method(s)	- Trainer() - Trainer(<i>parameters</i>) - getRegisterDate() - getExperience() - setData()	- calcComm() - calcSalary() - Elaun() - displayLevel() - toString()
Other classes: Table		
Method	-table()	
Other classes: Date		
Attributes	-day -month -year	
Method(s)	-Date(<i>paremeters</i>) -displayDate()	
Test Class: Main		
Content(s)	-array declaration -usage of I/O File System -usage of seven basic algorithm =sum =average =count =minimum =maximum =searching =sorting	

Class Design

(Class Table)

```
public class Table{
    public String table()
    {
        return
            "\n|-----|
            | " +
            "\n|   LEVELS   |   SPORTS   |   SESSIONS   |   PRICE
            | " +
            "\n|-----|
            | " +
            "\n|           |           |   Day       |
RM120.00 | " +
            "\n|           |   Taekwondo |-----|
            | " +
            "\n|           |           |   Night     |
RM180.00 | " +
            "\n|           |-----|
            | " +
            "\n|           |           |   Day       |
RM110.00 | " +
            "\n|           |   Kendo     |-----|
            | " +
            "\n|           |           |   Night     |
RM170.00 | " +
            "\n|1- Beginner |-----|
            | " +
            "\n|           |           |   Day       |
RM100.00 | " +
            "\n|           |   Silat     |-----|
            | " +
            "\n|           |           |   Night     |
RM160.00 | " +
            "\n|           |-----|
            | " +
            "\n|           |           |   Day       |
RM200.00 | " +
            "\n|           |   Boxing    |-----|
            | " +
            "\n|           |           |   Night     |
RM260.00 | " +
            "\n|-----|
            | " +
            "\n|           |           |   Day       |
RM150.00 | " +
            "\n|           |   Taekwondo |-----|
            | " +
            "\n|           |           |   Night     |
RM220.00 | " +
            "\n|-----|
            | " +
```

```

        "\n|          | Day |
RM130.00 |" +
        "\n| Kendo |-----
----|" +
        "\n| Night |
RM200.00 |" +
        "\n| 2-Advanced |-----
----|" +
        "\n| Day |
RM120.00 |" +
        "\n| Silat |-----
----|" +
        "\n| Night |
RM190.00 |" +
        "\n|-----
----|" +
        "\n| Day |
RM230.00 |" +
        "\n| Boxing |-----
----|" +
        "\n| Night |
RM300.00 |" +
        "\n|-----
----|";

    }
}

```

(Class Date)

```

public class Date
{
    private int day;
    private int month;
    private int year;

    public Date(int d, int m,int y)
    {
        this.day = d;
        this.month = m;
        this.year = y;
    }

    public String displayDate()
    {

```

```

        return "\nRegistration Date: " + day + "/" + month +
"/" + year;
    }
}

```

(Class Trainee)

```

import java.text.DecimalFormat;

public class Trainee extends Person
{
    private Date registerDate;
    private String member;

    DecimalFormat df = new DecimalFormat("0.00");
    public Trainee()
    {
        super();
        member = null;
        age =0;
    }//deafult cons

    public Trainee(String n,String i,String p,String e,String
add,String s,String ss,String id,int l,int a,String m)
    {
        super(n,i,p,e,add,s,ss,id,l,a);
        this.member = m;
    }//normal cons

    //acessor
    public Date getRegisterDate(){return registerDate;}
    public String getMember(){return member;}

    //mutator

```

```
public void setData(String newM)
{
    member = newM;
} //mutator

double price;
public double selectionPrice()
{

    if(level == 1)
    {
        if(sport.equalsIgnoreCase("Taekwondo"))
        {
            if(session.equalsIgnoreCase("Day"))
            {
                price = 120.00;
            }
            else if (session.equalsIgnoreCase("Night"))
            {
                price = 180.00;
            }
        }
        else if(sport.equalsIgnoreCase("Kendo"))
        {
            if(session.equalsIgnoreCase("Day"))
            {
                price = 110.00;
            }
            else if (session.equalsIgnoreCase("Night"))
            {
                price = 170.00;
            }
        }
        else if (sport.equalsIgnoreCase("Silat"))
```

```

{
    if(session.equalsIgnoreCase("Day"))
    {
        price = 100.00;
    }
    else if (session.equalsIgnoreCase("Night"))
    {
        price = 160.00;
    }
}
else if (sport.equalsIgnoreCase("Boxing"))
{
    if(session.equalsIgnoreCase("Day"))
    {
        price = 200.00;
    }
    else if (session.equalsIgnoreCase("Night"))
    {
        price = 260.00;
    }
}
}
else if(level == 2)
{
    if(sport.equalsIgnoreCase("Taekwondo"))
    {
        if(session.equalsIgnoreCase("Day"))
        {
            price = 150.00;
        }
        else if (session.equalsIgnoreCase("Night"))
        {
            price = 220.00;
        }
    }
}

```



```
}  
else if(sport.equalsIgnoreCase("Kendo"))  
{  
    if(session.equalsIgnoreCase("Day"))  
    {  
        price = 130.00;  
    }  
    else if (session.equalsIgnoreCase("Night"))  
    {  
        price = 200.00;  
    }  
}  
else if (sport.equalsIgnoreCase("Silat"))  
{  
    if(session.equalsIgnoreCase("Day"))  
    {  
        price = 120.00;  
    }  
    else if (session.equalsIgnoreCase("Night"))  
    {  
        price = 190.00;  
    }  
}  
else if (sport.equalsIgnoreCase("Boxing"))  
{  
    if(session.equalsIgnoreCase("Day"))  
    {  
        price = 230.00;  
    }  
    else if (session.equalsIgnoreCase("Night"))  
    {  
        price = 300.00;  
    }  
}
```

```

    }
    return price;
}

double totalPrice;
public double calculatePriceAfterDiscount()
{
    if(member.equalsIgnoreCase("Yes"))
    {
        totalPrice = price - (price * 0.10);
    }
    else if (member.equalsIgnoreCase("No"))
    {
        totalPrice = price - (price * 0.00);
    }

    return totalPrice;
}

public double Elaun()
{
    double elaun =0;
    if(age >= 18)
    {
        elaun = 500;
    }
    else
    {
        elaun = 0.00;
    }
    return elaun;
}

String lvl;

```

```

public String displayLevel()
{
    if(level == 1)
        lvl = "Beginner";
    else if (level == 2)
        lvl = "Advanced";

    return lvl;
}

//display
public String toString()
{
    return  super.toString() +
            "\nLevel : " + displayLevel() +
            "\nMembership : " + member +
            "\nPrice : RM " + df.format(selectionPrice())
+
            "\nTotal Price : RM " +
df.format(calculatePriceAfterDiscount()) +
            "\nTotal Elaun: RM " + df.format(Elaun());
}

}

```

(Class Trainer)

```

import java.text.DecimalFormat;

public class Trainer extends Person
{
    private Date registerDate;
    private String member;

```

```

private int expYears;

private final double basic = 2000.00;

    DecimalFormat df = new DecimalFormat("0.00");

    public Trainer()
    {
        super();
        expYears = 0;
    } //default cons

    public Trainer(String n,String i,String p,String
e,String add,String s, String ss,String id,int l,int a,String
m,int y)
    {
        super(n,i,p,e,add,s,ss,id,l,a);
        this.member = m;
        this.expYears = y;
    } //normal cons

    //acessor
    public Date getRegisterDate(){return registerDate;}
    public int getExperience() {return expYears;}

    //mutator
    public void setData(int newYear )
    {
        expYears = newYear;
    }

    double commission;
    public double calcComm()
    {
        if (level == 1)
        {
            if (session.equalsIgnoreCase("Day"))

```

```

        {
            commission = 0.05 * basic;
        }
        else if(session.equalsIgnoreCase("Night"))
        {
            commission = 0.10 * basic;
        }
    }
    else if (level == 2)
    {
        if (session.equalsIgnoreCase("Day"))
        {
            commission = 0.15 * basic;
        }
        else if(session.equalsIgnoreCase("Night"))
        {
            commission = 0.20 * basic;
        }
    }

    return commission;
}

```

```

double salary;
public double calcSalary()
{
    salary = basic + commission;

    return salary;
}

```

```

public double Elaun()
{

```

```

        double elaun =0;
        if(expYears >= 10)
        {
            elaun = 1000;
        }
        else
        {
            elaun = 0.00;
        }
        return elaun;
    }

    String lvl;
    public String displayLevel()
    {
        if(level == 1)
            lvl = "Beginner";
        else if (level == 2)
            lvl = "Advanced";

        return lvl;
    }

    //display
    public String toString()
    {
        return super.toString() +
            "\nLevel : " + displayLevel() +
            "\nExperience : " + expYears + " years" +
            "\nCommission : RM " + df.format(calcComm()) +
            "\nTotal Salary: RM " +
df.format(calcSalary()) +
            "\nTotal Elaun: RM " + df.format(Elaun());
    }

```

```

        public static String displayReport()
        {
            return "\n";
        }
    }
}

```

Class Main

```

import java.text.DecimalFormat;
import java.io.*;
import java.util.StringTokenizer;
import java.util.Scanner;
import javax.swing.*;

public class Main
{
    public static void main(String x[])
    {
        Scanner in = new Scanner (System.in);
        in.useDelimiter ("\n");

        try
        {

            //other declaration
            DecimalFormat df = new DecimalFormat("0.00");
            int i = 0;
            int j = 0;
            int n = 15;
            int E = 0;
            int k=0;
            int l=0;

```

```

//array decalration
Person Data[] = new Person[n];
Date[] dData = new Date[n];

//text file purposes
BufferedReader inPerson = new BufferedReader (new
FileReader ("Person.txt" ) );

PrintWriter outPerson = new PrintWriter (new
BufferedWriter(new FileWriter("Receipt.txt")));

String inData = null;

//intro
outPerson.print("\n");
outPerson.print("\t  ===== SELF-DEFENSE SPORTS CENTRE
=====");

outPerson.print("\n");

Table t = new Table();
outPerson.print(t.table()); //call table

outPerson.print("\n\n");

//person
while((inData = inPerson.readLine()) != null)
{
    StringTokenizer st = new
StringTokenizer(inData, ";");

    String name = st.nextToken();
    String ic = st.nextToken();
    String phone = st.nextToken();
    String email = st.nextToken();
    String address = st.nextToken();
    String sport = st.nextToken();

```



```

        String session = st.nextToken();
        int level = Integer.parseInt(st.nextToken());

        int option = Integer.parseInt(st.nextToken());
        if(option == 1)
        {
            int day =
Integer.parseInt(st.nextToken());

            int month =
Integer.parseInt(st.nextToken());

            int year =
Integer.parseInt(st.nextToken());

            dData[i] = new Date(day,month,year);

            String member = st.nextToken();

            int age =
Integer.parseInt(st.nextToken());

            String id = "TE00" + (k+1);

            Data[i] = new
Trainee(name,ic,phone,email,address,sport,session,id,level,age
,member);

            k++;
        }
        else if(option == 2)
        {
            int day =
Integer.parseInt(st.nextToken());

            int month =
Integer.parseInt(st.nextToken());

            int year =
Integer.parseInt(st.nextToken());

            dData[i] = new Date(day,month,year);

            String member = st.nextToken();

            int expYears =
Integer.parseInt(st.nextToken());

            int age =
Integer.parseInt(st.nextToken());

```

```

        String id = "TR00" + (l+1);

        Data[i] = new
Trainer(name,ic,phone,email,address,sport,session,id,level,age
,member,expYears);

        l++;
    }
    i++;
}

outPerson.println();
outPerson.println("<<  List of Trainee  >>");
outPerson.print("-----");
for( i=0; i < Data.length; i++)
{
    //output file version
    outPerson.print("\n\n");
    if(Data[i] instanceof Trainee)
    {
        outPerson.println(Data[i].toString());
        outPerson.println(dData[i].displayDate());
    }
}

outPerson.println("\n\n<<  List of Trainer  >>");
outPerson.print("-----");
for( j=0; j< Data.length; j++)
{
    //output file version
    outPerson.print("\n\n");
    if(Data[j] instanceof Trainer)
    {
        outPerson.println(Data[j].toString());
    }
}

```

```

        outPerson.println(dData[j].displayDate());
    }
}

double totalIncome=0;
//total income(Trainee)
for(i=0;i<Data.length;i++)
{
    if(Data[i] instanceof Trainee)
    {
        totalIncome = totalIncome +
((Trainee)Data[i]).calculatePriceAfterDiscount();
    }
}
//average income(Trainee)
double avg1 = totalIncome/k;

double totalSalary=0;
//total salary(Trainer)
for(i=0;i<Data.length;i++)
{
    if(Data[i] instanceof Trainer)
    {
        totalSalary = totalSalary+
((Trainer)Data[i]).calcSalary();
    }
}
//average salary (Trainer)
double avg2 = totalSalary/l;

//highest paid fee
double high = 0;
for( i=0; i < Data.length; i++)

```

```

        {
            if(Data[i] instanceof Trainee)
            {
                if(high <
((Trainee)Data[i]).calculatePriceAfterDiscount())
                {
                    high =
((Trainee)Data[i]).calculatePriceAfterDiscount();
                }
            }
        }

//lowest paid salary
double low = 9999999;
for( i=0; i < Data.length; i++)
{
    if(Data[i] instanceof Trainer)
    {
        if(low > ((Trainer)Data[i]).calcSalary())
        {
            low = ((Trainer)Data[i]).calcSalary();
        }
    }
}

//count Trainee fee < 200.00
int countTe = 0;
for( i=0; i < Data.length; i++)
{
    if(Data[i] instanceof Trainee)
    {
        if(((Trainee)Data[i]).calculatePriceAfterDiscount() < 200.00)
        {
            countTe++;
        }
    }
}

```

```

    }
}

//count Trainer salary > 2100.00
int countTr = 0;
for( i=0; i < Data.length; i++)
{
    if(Data[i] instanceof Trainer)
    {
        if(((Trainer)Data[i]).calcSalary() > 2100)
        {
            countTr++;
        }
    }
}

```

```

System.out.println("COMPILING SUCCESSFUL! - NO SYNTAX
ERRORS");

```

```

//monthly report
//output file version(TRAINEE)
outPerson.print("\n\n");
outPerson.print("\n===TRAINEE REPORT===");
outPerson.print("\n\nTotal Fee: RM " +
df.format(totalIncome));
outPerson.print("\nAverage Fee: RM " +
df.format(avgl));
outPerson.print("\nThe Highest Total Fee in
Trainee: RM " + df.format(high));
outPerson.print("\nThe Total Trainee with fee
less than RM200: " + countTe + " trainees");

```

```

//output file version(TRAINER)
outPerson.print("\n\n");
outPerson.print("\n===TRAINER REPORT===");
        outPerson.print("\n\nTotal Salary Given: RM " +
df.format(totalSalary));
        outPerson.print("\nAverage Salary Given: RM " +
df.format(avg2));
        outPerson.println("\nThe Lowest Paid Salary in
Trainer: RM " + df.format(low));
        outPerson.println("The Total Trainer with salary
above RM2100: " + countTr + " trainers");

```

```

//sorting age youngest to oldest
int a;
int b;
Person temp;

for(a = 0; a<Data.length; a++)
{
    for(b=a+1; b<Data.length;b++)
    {
        if(Data[a].getAge() > Data[b].getAge())
        {
            temp = Data[a];
            Data[a] = Data[b];
            Data[b] = temp;
        }
    }
}

```

```

        outPerson.println("\n\n\n---Sorting Age of Person
(Youngest to Oldest)---\n");
        for( a=0; a<Data.length; a++)
        {
            outPerson.println(Data[a].getName() + " - " +
Data[a].getAge() + " years old");

```

```

    }

    String sid = null;
    String mode = null;
    int loc=0;
    boolean found = false;

    System.out.println("\n\nCHOOSE OPTION:\n[1] -
    SEARCH\n[2] - EXIT ");
    int option = in.nextInt();
    if(option == 1)
    {
        mode = JOptionPane.showInputDialog("ENTER
        MODE:\n[1]-TRAINEE\n[2]-TRAINER");
        if(mode.equalsIgnoreCase("1"))
        {
            sid = JOptionPane.showInputDialog("\n===TRAINEE
            ID
            LIST===\n[TE001]\n[TE002]\n[TE003]\n[TE004]\n[TE005]\n[TE006]\n
            [TE007]\n[TE008]\n[TE009]\n\nSearch Trainee ID : ");
            while(loc<=Data.length && !found)
            {
                if(Data[loc].getuniqueID().equalsIgnoreCase(sid))
                {
                    found=true;
                }
                else { loc++; }
            }

            if(Data[loc] instanceof Trainee)
            {
                if(found)

```

```

        {

JOptionPane.showMessageDialog(null,"===PERSON'S INFO===" +
"\n" + Data[loc].toString());

        }
        else
        {

JOptionPane.showMessageDialog(null,"PERSON NOT FOUND!");

        }
    }
    else if(mode.equalsIgnoreCase("2"))
    {
        sid = JOptionPane.showInputDialog("\n===TRAINER
ID
LIST===\n[TR001]\n[TR002]\n[TR003]\n[TR004]\n[TR005]\n[TR006]\n
\nSearch Trainer ID : ");
        while(loc<=Data.length && !found)
        {

if(Data[loc].getuniqueID().equalsIgnoreCase(sid))
        {
            found=true;
        }
        else { loc++; }
    }

    if(Data[loc] instanceof Trainer)
    {
        if(found)
        {

JOptionPane.showMessageDialog(null,"===PERSON'S INFO===" +
"\n" + Data[loc].toString());

        }
        else
        {

```



```

JOptionPane.showMessageDialog(null,"PERSON NOT FOUND!");
        }
    }
}
else
{
    JOptionPane.showMessageDialog(null, "\nTHANK
YOU!");
}

//text file purposes
inPerson.close();
outPerson.close();
} //end of try

catch (FileNotFoundException fnfe)
{
    System.err.println("File Not Found !");
}
catch (IOException io)
{
    System.out.println (io.getMessage());
}
catch (Exception e)
{
    System.out.println (e.getMessage());
}
} //void main
} //end of class

```

SAMPLE INPUT AND OUTPUT

Input File

Person - Notepad

File Edit Format View Help

```
Midoriya Izuku;030715-00-1234;+8112345678;dekun@gmail.com;17th A, Shizuoka Street;Boxing;Night;1;14;01;2019;No;18
Yagi Toshinori;800610-01-8612;+8123853274;almight@gmail.com;88th A, Shizuoka Street;Boxing;Night;2;2;01;2019;Yes;20;41
Bakugo Katsuki;030420-00-5678;+8198765423;kacchan@gmail.com;17 Aizawa Shota;Taekwondo;Night;2;1;15;01;2019;Yes;17
Aizawa Shota;801108-01-1234;+8196857412;eraserhead@gmail.com;02 Section A, Shizune Street;Taekwondo;Day;1;2;01;01;2014;No;8;34
Uraraka Ochocho;031227-00-5298;+8178965413;uravityn@gmail.com;20th C, Hikari Street;Boxing;Day;1;1;20;02;2019;No;16
Yamada Hizashi;800707-01-9654;+8123698456;yamadasan@gmail.com;Section B, Minori Street;Kendo;Night;2;2;01;01;2015;Yes;15;40
Iida Tenya;030822-00-7856;+8185631232;crIida@gmail.com;27th A, Matsumoto Street;Taekwondo;Day;2;1;19;02;2019;Yes;20
Ishiyama Ken;850301-01-6574;+863142523;kenkun@gmail.com;44th C, Matsumoto Street;Silat;Night;1;2;05;01;2015;No;9;38
Asui Tsuyu;030212-00-2546;+81851232964;fropopy@gmail.com;02 Section A, Shizune Street;Silat;Night;1;1;12;03;2019;Yes;18
Kayama Nemuri;871201-01-1268;+836951247;midnightsama@gmail.com;20th C, Hikari Street;Kendo;Day;1;2;02;01;2016;No;10;36
Yaoyorozu Momo;030923-00-3366;+8165874120;creati@gmail.com;Avenue Mansion,Aichi Street;Taekwondo;Night;1;1;30;01;2019;Yes;22
Torino Sorahiko;700130-01-9621;+836214596;grandtorino@gmail.com;30th F, Ai Street;Boxing;Night;1;2;18;06;2000;Yes;17;66
Todoroki Shoto;030111-00-7755;+8198563214;flamingice@gmail.com;Todoroki Mansion, Shizuoka Street;Kendo;Night;2;1;30;01;2019;Yes;16
Ashido Mina;030730-00-983274;+8178563241;pinkyqueen@gmail.com;1st C, Minori Street;Silat;Night;1;1;12;03;2019;No;25
Kaminari Denki;030618-00-6584;+8132984563;stunGuNari@gmail.com;30th F, Ai Street;Silat;Night;1;1;22;02;2019;Yes;20
```

Output File

Receipt - Notepad

File Edit Format View Help

<< List of Trainee >>

Name : Midoriya Izuku
IC Number : 030715-00-1234
Phone : +8112345678
Email : dekun@gmail.com
Area Code : 17th A, Shizuoka Street
ID : TE001
Sport : Boxing
Session : Night
Age : 18 years old
Level : Beginner
Membership : No
Price : RM 260.00
Total Price : RM 260.00
Total Elaun: RM 500.00

Registration Date: 14/1/2019

Receipt - Notepad

File Edit Format View Help

Name : Uraraka Ochocho
IC Number : 031227-00-5298
Phone : +8178965413
Email : uravityn@gmail.com
Area Code : 20th C, Hikari Street
ID : TE003
Sport : Boxing
Session : Day
Age : 16 years old
Level : Beginner
Membership : No
Price : RM 200.00
Total Price : RM 200.00
Total Elaun: RM 0.00

Registration Date: 20/2/2019

Name : Bakugo Katsuki
IC Number : 030420-00-5678
Phone : +8198765423
Email : kacchan@gmail.com
Area Code : 17 Aizawa Shota
ID : TE002
Sport : Taekwondo
Session : Night
Age : 17 years old
Level : Advanced
Membership : Yes
Price : RM 220.00
Total Price : RM 198.00
Total Elaun: RM 0.00

Registration Date: 15/1/2019

Name : Iida Tenya
IC Number : 030822-00-7856
Phone : +8185631232
Email : crIida@gmail.com
Area Code : 27th A, Matsumoto Street
ID : TE004
Sport : Taekwondo
Session : Day
Age : 20 years old
Level : Advanced
Membership : Yes
Price : RM 150.00
Total Price : RM 135.00
Total Elaun: RM 500.00


Registration Date: 19/2/2019

 Receipt - Notepad
File Edit Format View Help
Name : Asui Tsuyu
IC Number : 030212-00-2546
Phone : +81851232964
Email : froppy@gmail.com
Area Code : 02 Section A, Shizune Street
ID : TE005
Sport : Silat
Session : Night
Age : 18 years old
Level : Beginner
Membership : Yes
Price : RM 160.00
Total Price : RM 144.00
Total Elaun: RM 500.00


Registration Date: 12/3/2019

Name : Yaoyorozu Momo
IC Number : 030923-00-3366
Phone : +8165874120
Email : creati@gmail.com
Area Code : Avenue Mansion,Aichi Street
ID : TE006
Sport : Taekwondo
Session : Night
Age : 22 years old
Level : Beginner
Membership : Yes
Price : RM 180.00
Total Price : RM 162.00
Total Elaun: RM 500.00

Registration Date: 30/1/2019

 Receipt - Notepad
File Edit Format View Help
Name : Kaminari Denki
IC Number : 030618-00-6584
Phone : +8132984563
Email : stunGuNari@gmail.com
Area Code : 30th F, Ai Street
ID : TE009
Sport : Silat
Session : Night
Age : 20 years old
Level : Beginner
Membership : Yes
Price : RM 160.00
Total Price : RM 144.00
Total Elaun: RM 500.00

Registration Date: 22/2/2019

 Receipt - Notepad
File Edit Format View Help
Name : Todoroki Shoto
IC Number : 030111-00-7755
Phone : +8198563214
Email : flamingice@gmail.com
Area Code : Todoroki Mansion, Shizuoka Street
ID : TE007
Sport : Kendo
Session : Night
Age : 16 years old
Level : Advanced
Membership : Yes
Price : RM 200.00
Total Price : RM 180.00
Total Elaun: RM 0.00

Registration Date: 30/1/2019

Name : Ashido Mina
IC Number : 030730-00-983274
Phone : +8178563241
Email : pinkyqueen@gmail.com
Area Code : 1st C, Minori Street
ID : TE008
Sport : Silat
Session : Night
Age : 25 years old
Level : Beginner
Membership : No
Price : RM 160.00
Total Price : RM 160.00
Total Elaun: RM 500.00

Registration Date: 12/3/2019

Receipt - Notepad

File Edit Format View Help

<< List of Trainer >>

Name : Yagi Toshinori
IC Number : 800610-01-8612
Phone : +8123853274
Email : almight@gmail.com
Area Code : 88th A, Shizuoka Street
ID : TR001
Sport : Boxing
Session : Night
Age : 41 years old
Level : Advanced
Experience : 20 years
Commission : RM 400.00
Total Salary: RM 2400.00
Total Elaun: RM 1000.00

Registration Date: 1/1/2019

Name : Aizawa Shota
IC Number : 801108-01-1234
Phone : +8196857412
Email : eraserhead@gmail.com
Area Code : 02 Section A, Shizune Street
ID : TR002
Sport : Taekwondo
Session : Day
Age : 34 years old
Level : Beginner
Experience : 8 years
Commission : RM 100.00
Total Salary: RM 2100.00
Total Elaun: RM 0.00

Name : Torino Sorahiko
IC Number : 700130-01-9621
Phone : +836214596
Email : grandtorino@gmail.com
Area Code : 30th F, Ai Street
ID : TR006
Sport : Boxing
Session : Night
Age : 66 years old
Level : Beginner
Experience : 17 years
Commission : RM 200.00
Total Salary: RM 2200.00
Total Elaun: RM 1000.00

Registration Date: 18/6/2000

Receipt - Notepad

File Edit Format View Help

Name : Yamada Hizashi
IC Number : 800707-01-9654
Phone : +8123698456
Email : yamadasan@gmail.com
Area Code : Section B, Minori Street
ID : TR003
Sport : Kendo
Session : Night
Age : 40 years old
Level : Advanced
Experience : 15 years
Commission : RM 400.00
Total Salary: RM 2400.00
Total Elaun: RM 1000.00


Registration Date: 1/1/2015

Name : Ishiyama Ken
IC Number : 850301-01-6574
Phone : +863142523
Email : kenkun@gmail.com
Area Code : 44th C, Matsumoto Street
ID : TR004
Sport : Silat
Session : Night
Age : 38 years old
Level : Beginner
Experience : 9 years
Commission : RM 200.00
Total Salary: RM 2200.00
Total Elaun: RM 0.00

Registration Date: 5/1/2015

Name : Kayama Nemuri
IC Number : 871201-01-1268
Phone : +836951247
Email : midnightsama@gmail.com
Area Code : 20th C, Hikari Street
ID : TR005
Sport : Kendo
Session : Day
Age : 36 years old
Level : Beginner
Experience : 10 years
Commission : RM 100.00
Total Salary: RM 2100.00
Total Elaun: RM 1000.00

Registration Date: 2/1/2016

 *Receipt - Notepad

File Edit Format View Help

====TRAINEE REPORT====

Total Fee: RM 1583.00
Average Fee: RM 175.89
The Highest Total Fee in Trainee: RM 260.00
The Total Trainee with fee less than RM200: 7 trainees

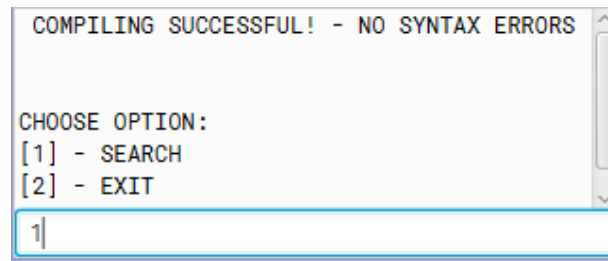
====TRAINER REPORT====

Total Salary Given: RM 13400.00
Average Salary Given: RM 2233.33
The Lowest Paid Salary in Trainer: RM 2100.00
The Total Trainer with salary above RM2100: 4 trainers

---Sorting Age of Person (Youngest to Oldest)---

Uraraka Ocacho - 16 years old
Todoroki Shoto - 16 years old
Bakugo Katsuki - 17 years old
Asui Tsuyu - 18 years old
Midoriya Izuku - 18 years old
Iida Tenya - 20 years old
Kaminari Denki - 20 years old
Yaoyorozu Momo - 22 years old
Ashido Mina - 25 years old
Aizawa Shota - 34 years old
Kayama Nemuri - 36 years old
Ishiyama Ken - 38 years old
Yamada Hizashi - 40 years old
Yagi Toshinori - 41 years old
Torino Sorahiko - 66 years old

Input / Output Searching



COMPILING SUCCESSFUL! - NO SYNTAX ERRORS

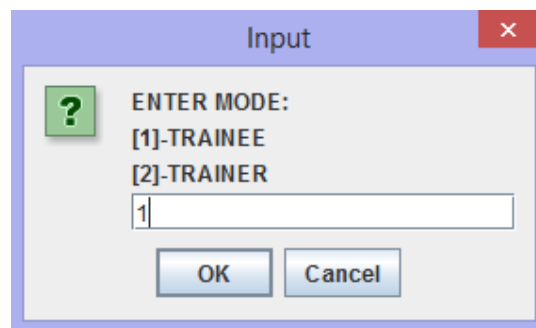
CHOOSE OPTION:

[1] - SEARCH

[2] - EXIT

1

Option 1: Search from Trainee



Input

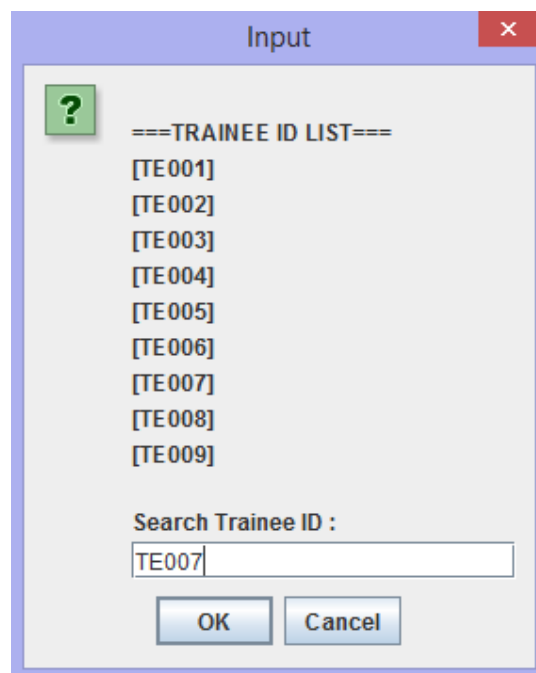
ENTER MODE:

[1]-TRAINEE

[2]-TRAINER

1

OK Cancel



Input

===TRAINEE ID LIST===

[TE001]

[TE002]

[TE003]

[TE004]

[TE005]

[TE006]

[TE007]

[TE008]

[TE009]

Search Trainee ID :

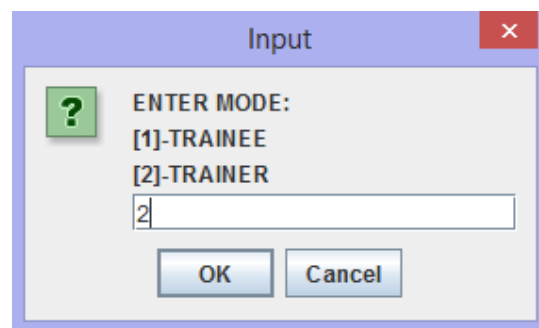
TE007

OK Cancel

Ouput:



Option 2: Search from Trainers



Input

?

===TRAINER ID LIST===

[TR001]
[TR002]
[TR003]
[TR004]
[TR005]
[TR006]

Search Trainer ID :

TR002

OK Cancel

Output:

Message

i

===PERSON'S INFO===

Name : Aizawa Shota
IC Number : 801108-01-1234
Phone : +8196857412
Email : eraserhead@gmail.com
Area Code : 02 Section A, Shizune Street
ID : TR002
Sport : Taekwondo
Session : Day
Age : 34 years old
Level : Beginner
Experience : 8 years
Commission : RM 100.00
Total Salary: RM 2100.00
Total Elaun: RM 0.00

OK

Rubric Assignment 4 (Total marks: 59)

Tasks		0	1	2	Weight	Total
		Not exist	Exist			
			Incorrect/ incomplete	Complete		
Superclass	Default constructor				1	
	Normal constructor				1	
	Mutator				1	
	Accessors				1	
	Processor method				2	
	toString()				1	
Subclasses	Default constructor				1	
	Normal constructor				1	
	Mutator				1	
	Accessors				1	
	Processor method				2	
	toString()				1	
Main/Test class	Array declaration				1	
	User input				2	
	Object creation				2	
	Display the information of the objects and output from processor method.				3	
Overall execution		1	2	3	3	
		Poor	Crash in some areas	Executed perfectly		
Submission		Very late without reasonable excuse	Late with acceptable reason	Right on time	2	
TOTAL (59 MARKS)						

Rubric Final Project (Total marks: 108)

Final Project Rubric					
Proposal					
Tasks	1	2	3	Weight	Total
Class selection	Not an appropriate class selection	Good class selection but too simple without careful thought	Good selection and well thought	3	
Attributes	Attributes are not appropriate	Appropriate but incomplete	Good and appropriate attributes	3	
Methods	Not an appropriate method selection	Good method selection but too simple without careful thought	Good selection and well thought	3	
Submission	Very late without reasonable excuse	Late with acceptable reason	Right on time	2	
Class Design					
Tasks	0	1	2	Weight	Total
Superclass Constructor	Not exist	Exist but incomplete	Complete	2	
Subclasses Constructor	Not exist	Exist but incomplete	Complete	2	
Accessors	Not exist	Exist but incomplete	Complete	1	
Mutators	Not exist	Exist but incomplete	Complete	1	
toString()	Not exist	Exist but incomplete	Complete	1	
Object					
Creation	Fail to create object	Incomplete object creation	Object created successfully	2	
Method Call	Fail to show method call for any subclass	There are method call of a subclass but with error	Complete and successful	3	

Method definition	No abstract method defined	There is an abstract method but incomplete definition	Completely defined for each subclass	3	
Method call	Fail to show method call for abstract method	Successfully called but unnecessarily with casting and instanceof	Called correctly	3	
Overall					
Tasks	1	2	3	Weight	Total
Overall Execution	Program failed to execute	Executed with inaccurate output	Executed and output displayed correctly	3	
Superclass and subclasses selection	Not an appropriate class selection	Good class selection but too simple without careful thought	Good selection and properly implemented and well thought.	2	
Timeliness	Very late without reasonable excuse	Late with acceptable reason	Right on time	3	
Completeness	Not compliance with the project description	Comply with project description	Exceed project's expectation	2	
Report	Did not follow the guideline given.	Format and content 70% according to the guideline given.	Format and content according to the guideline given.	3	
TOTAL MARKS					108