

**CSC186 – OBJECT ORIENTED**

**PROGRAMMING**

**LAB ASSIGNMENT 2**

NAME : MUHAMMAD REDZA BIN MAHAYADIN

STUDENT ID : 2022676696

GROUP : RCDCS1102B

LECTURER : SIR MOHD NIZAM BIN OSMAN

SOURCE CODE 1.1 : PROGRAM CLASS

public class Program {

    private String progCode;

    private String progDesc;

    private String duration;

    private String faculty;

    private String progHead;

    //default constructor

    public Program()

    {

        progCode = "";

        progDesc = "";

        duration = "";

        faculty = "";

        progHead = "";

    }

    //normal constructor

    public Program(String pc, String pd, String dr, String fc, String ph)

    {

        progCode = pc;

        progDesc = pd;

        duration = dr;

        faculty = fc;

        progHead = ph;

    }

    //copy constructor

    public Program(Program copyProgram)

    {

        this.progCode = copyProgram.progCode;

        this.progDesc = copyProgram.progDesc;

        this.duration = copyProgram.duration;

        this.faculty = copyProgram.faculty;

        this.progHead = copyProgram.progHead;

    }

    //setter

    public void setProgCode(String pc)

    {

        progCode = pc;

    }

    public void setProgDesc(String pd)

    {

        progDesc = pd;

    }

    public void setDuration(String dr)

    {

        duration = dr;

    }

    public void setFaculty(String fc)

    {

        faculty = fc;

    }

    public void setProgHead(String ph)

    {

        progHead = ph;

    }

    //getter

    public String getProgCode()

    {

        return progCode;

    }

    public String getProgDesc()

    {

        return progDesc;

    }

    public String getDuration()

    {

        return duration;

    }

    public String getFaculty()

    {

        return faculty;

    }

    public String getProgHead()

    {

        return progHead;

    }

    //processor

    public String programLevel()

    {

        char progCode = getProgCode().charAt(2);

        if(progCode=='0')

            return "Certificate";

        else if(progCode=='1')

            return "Diploma";

        else if(progCode=='2')

            return "Degree";

        else if(progCode=='7')

            return "Master";

        else if(progCode=='9')

            return "PhD";

        else

            return "Invalid Program Code.";

    }

    //printer

    public String toString()

    {

        return String.format("%nProgram Code: %s%nProgram Description: %s%nProgram Level: %s%nDuration: %s%nFaculty: %s%nProgram Head: %s%n", progCode, progDesc, programLevel(), duration, faculty, progHead);

    }

}

SOURCE CODE 1.2 : LAND CLASS

public class Land {

    private String id;

    private String ownerName;

    private String houseType;

    private double area;

    //default constructor

    public Land()

    {

        id = "";

        ownerName = "";

        houseType = "";

        area = 0;

    }

    //normal constructor

    public Land(String id, String ownerName, String houseType, double area)

    {

        this.id = id;

        this.ownerName = ownerName;

        this.houseType = houseType;

        this.area = area;

    }

    //copy constructor

    public Land(Land copyLand)

    {

        this.id = copyLand.id;

        this.ownerName = copyLand.ownerName;

        this.houseType = copyLand.houseType;

        this.area = copyLand.area;

    }

    //getter

    public String getId()

    {

        return this.houseType;

    }

    public String getOwnerName()

    {

        return this.ownerName;

    }

    public String getHouseType()

    {

        return this.houseType;

    }

    public double getArea()

    {

        return this.area;

    }

    //processor

    public double calcTaxPrice()

    {

        double taxRate = 0, taxPrice = 0; // RM-per-m^2

        switch (getHouseType().charAt(0)) {

            case 't':

            case 'T':

                taxRate = 10;

                break;

            case 's':

            case 'S':

                taxRate = 15;

                break;

            case 'b':

            case 'B':

                taxRate = 20;

                break;

            case 'c':

            case 'C':

                taxRate = 30;

                break;

            default:

                System.out.println("Invalid house type.");

                break;

        }

        taxPrice = taxRate\*getArea();

        return taxPrice;

    }

    //printer

    public String toString()

    {

        return String.format("%nLand ID:\t%s%nOwner Name:\t%s%nHouse Type:\t%s%nArea of land:\t%.2f m^2%nTax Price:\tRM%.2f%n", id, ownerName, houseType, area, calcTaxPrice());

    }

}

SOURCE CODE 1.3 : MAIN CLASS

import java.util.Scanner;;

public class Main {

    public static void main(String[] args)

    {

        int choice = selectProgram();

        if(choice == 1 )

        {

            //instantiate an object

            Program p = new Program();

            //input data

            Scanner in = new Scanner(System.in);

            System.out.print("\nEnter program code: ");

            String pc = in.nextLine();

            System.out.print("Enter program description: ");

            String pd = in.nextLine();

            System.out.print("Enter duration: ");

            String dr = in.nextLine();

            System.out.print("Enter faculty: ");

            String fc = in.nextLine();

            System.out.print("Enter program head: ");

            String ph = in.nextLine();

            in.close();

            //store onto object

            //normal constructor

            p = new Program(pc, pd, dr, fc, ph);

            //OR

            //Setter

            p.setProgCode(pc);

            p.setProgDesc(pd);

            p.setDuration(dr);

            p.setFaculty(fc);

            p.setProgHead(ph);

            //manipulate data

            System.out.println(p.toString());

        }

        else if(choice == 2)

        {

            Land l = new Land();

            Scanner in = new Scanner(System.in);

            System.out.print("\nEnter land id: ");

            String id = in.nextLine();

            System.out.print("Enter owner name: ");

            String ownerName = in.nextLine();

                // capitalize each of the first letter of the owner name

                for (int i = 0; i < ownerName.length(); i++)

                {

                    if (i == 0) {

                        ownerName = ownerName.substring(0, 1).toUpperCase() + ownerName.substring(1);

                    } else if (ownerName.charAt(i) == ' ') {

                        ownerName = ownerName.substring(0, i + 1) + ownerName.substring(i + 1, i + 2).toUpperCase()

                                + ownerName.substring(i + 2);

                    }

                }

            System.out.print("Enter house type [T,S,B,C]: ");

            String houseType = in.nextLine();

                switch (houseType.charAt(0)) {

                    case 't':

                    case 'T':

                        houseType = "Terrace";

                        break;

                    case 's':

                    case 'S':

                        houseType = "Semi-Detached";

                        break;

                    case 'b':

                    case 'B':

                        houseType = "Bungalow";

                        break;

                    case 'c':

                    case 'C':

                        houseType = "Condominium";

                        break;

                    default:

                        System.out.println("Invalid house type.");

                        System.exit(1);

                        break;

                }

            System.out.print("Enter area: ");

            double area = in.nextDouble();

            in.close();

            l = new Land(id, ownerName, houseType, area);

            System.out.println(l.toString());

        }

    }

    public static int selectProgram()

    {

        Scanner in = new Scanner(System.in);

        System.out.printf("%n1.University Program Level%n2.Land details%n%nSelect program[1-2]: ");

        int choice = in.nextInt();

            if(choice != 1 && choice != 2)

            {

                System.out.println("Invalid choice.");

                System.exit(1);

            }

        return choice;

    }

}

SAMPLE INPUT (PROGRAM)

A picture containing text, screenshot, font

Description automatically generated

SAMPLE OUTPUT (PROGRAM)

A black screen with white text

Description automatically generated with low confidence

SAMPLE INPUT (LAND)

A black background with white text

Description automatically generated with low confidence

SAMPLE OUTPUT (LAND)

A picture containing text, screenshot, font, black

Description automatically generated