

**CSC186 – OBJECT ORIENTED**

**PROGRAMMING**

**LAB ASSIGNMENT 5**

NAME : MUHAMMAD REDZA BIN MAHAYADIN

STUDENT ID : 2022676696

GROUP : RCDCS1102B

LECTURER : SIR MOHD NIZAM BIN OSMAN

QUESTION 6.1

SOURCE CODE 1.1 : FoodApp Class

import java.util.\*;

public class FoodApp {

    public static void main(String[]args) {

        Scanner scan = new Scanner(System.in);

        Scanner scan1 = new Scanner(System.in);

        //step 1: declare array of object

        System.out.println("enter size of array: ");

        int size = scan.nextInt();

        WesternFood[] wf = new WesternFood[size];

        //step 2: create / instantiate array of object

        for(int i = 0; i < size; i++){

            wf[i] = new WesternFood();

        }

        // Food set

        System.out.println();

        System.out.println("Set\tDescription\tPrice");

        System.out.println("1\tLamb Chop\tRM30.00");

        System.out.println("2\tChicken Chop\tRM20.00");

        System.out.println("3\tFish Chop\tRM15.00");

        //step 3: input

        for(int i = 0; i < size; i++){

            System.out.println();

            System.out.println("Enter details for customer " + (i+1));

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

            String name = scan1.nextLine();

            System.out.println("Enter quantity order: ");

            int quantity = scan.nextInt();

            System.out.println("Enter member [true/false]: ");

            boolean member = scan.nextBoolean();

            System.out.println("Enter food set [1-3]: ");

            int foodSet = scan.nextInt();

            System.out.println("Enter dessert [true/false]: ");

            boolean dessert = scan.nextBoolean();

            wf[i].setDessert(dessert);

            //step 4: store onto array

            //1. normal constructor

            wf[i] = new WesternFood(name, quantity, member, foodSet, dessert);

            //or

            //2.setter

            wf[i].setName(name);

            wf[i].setQuantity(quantity);

            wf[i].setMember(member);

            wf[i].setFoodSet(foodSet);

            wf[i].setDessert(dessert);

        }

        System.out.println();

        //step 5: manipulation

        //q(ii)

        for(int i = 0; i < size; i++){

            System.out.println();

            System.out.println(wf[i].toString() + "Payment: RM" + wf[i].Payment());

        }

        //q(iii)

        int cntDessert = 0;

        for(int i = 0; i < size; i++){

            if(wf[i].getDessert())

                cntDessert++;

        }

        System.out.println();

        System.out.println("Number of customers order dessert: " + cntDessert);

        //q(iv)

        double totalPriceMember = 0; //total price for member

        for(int i = 0; i < size; i++) {

            if(wf[i].getMember())

                totalPriceMember = totalPriceMember + wf[i].Payment();

        }

        System.out.println("Total payment for all members: RM" + totalPriceMember);

        //q(v)

        double totalPrice = 0;

        for(int i = 0; i < size; i++){

            totalPrice = totalPrice + wf[i].Payment();

        }

        System.out.println("Total payment for all customers: RM" + totalPrice);

        //q(vi)

        System.out.println();

        System.out.println("Customer who order the Lamb Chop set");

        for(int i = 0; i < size; i++){

            if(wf[i].getFoodSet() == 1){

                System.out.println();

                System.out.println(wf[i].toString() + "Payment: " + wf[i].Payment());

            }

        }

        scan.close();

        scan1.close();

    }//end main

} //end class

SOURCE CODE 1.2 : Food Class

public class Food{

    //Data members

    private String name;

    private int quantityOfOrder;

    private boolean member;

    //method members

    //default constructor

    public Food(){

        name = "";

        quantityOfOrder = 0;

        member = false;

    }

    //copy constructor

    public Food(Food f){

        name = f.name;

        quantityOfOrder = f.quantityOfOrder;

        member = f.member;

    }

    //normal constructor

    public Food(String name, int quantity, boolean member){

        this.name = name;

        this.quantityOfOrder = quantity;

        this.member = member;

    }

    //getter

    public String getName(){

        return name;

    }

    public double getQuantityOfOrder(){

        return quantityOfOrder;

    }

    public boolean getMember(){

        return member;

    }

    //setter

    public void setName(String name){

        this.name = name;

    }

    public void setQuantity(int quantity){

        this.quantityOfOrder = quantity;

    }

    public void setMember(boolean member){

        this.member = member;

    }

    //printer

    public String toString(){

        return "Name: " + name + "\nQuantity: " + quantityOfOrder + "\nMember: " + member;

    }

}

SOURCE CODE 1.3 : WesternFood Class

public class WesternFood extends Food {

    //data members

    private int foodSet;

    private boolean dessert;

    //method members

    //default constructors

    public WesternFood(){

        super();

        foodSet = 0;

        dessert = false;

    }

    //normal constructor

    public WesternFood(String name, int quantityOfOrder, boolean member, int foodSet, boolean dessert){

        super();

        this.foodSet = foodSet;

        this.dessert = dessert;

    }

    //setter

    public void setFoodSet(int foodSet){

        this.foodSet = foodSet;

    }

    public void setDessert(boolean dessert){

        this.dessert = dessert;

    }

    //getter

    public int getFoodSet(){

        return foodSet;

    }

    public boolean getDessert(){

        return dessert;

    }

    //processor

    public double Payment(){

        double price;

        if (foodSet == 1){

            price = 30.00;

        }

        else if (foodSet == 2){

            price = 20.00;

        }

        else if (foodSet == 3){

            price = 15.00;

        }

        else{

            price = 0.00;

        }

        double totalPrice = price \* getQuantityOfOrder();

        if(dessert)

            totalPrice = totalPrice + 10.90 \* getQuantityOfOrder();

        if(getMember())

            totalPrice = totalPrice \* 0.9;

        return totalPrice;

    }

    //printer

    public String toString(){

        return super.toString() + "\nFood Set: " + foodSet + "\nDessert: " + dessert + "\n";

    }

QUESTION 6.2

SOURCE CODE 2.1 : ArtistApp Class

import java.util.Scanner;

public class ArtistApp {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        Scanner in1 = new Scanner(System.in);

        Artist[] artists = new Artist[50];

        // Input data for each artist

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

            System.out.println();

            System.out.println("\tArtist " + (i + 1));

            System.out.println("Enter artist type (1 - Performer, 2 - Painter):");

            int artistProfession = in1.nextInt();

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

            String artistName = in.nextLine();

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

            double payment = in1.nextDouble();

            if (artistProfession == 1) {

                System.out.println("Enter working hours:");

                int workingHours = in1.nextInt();

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

                String genre = in.nextLine();

                artists[i] = new Performer(artistName, payment, workingHours, genre);

            } else if (artistProfession == 2) {

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

                int numOfPainting = in1.nextInt();

                artists[i] = new Painter(artistName, payment, numOfPainting);

            }

        }

        // Display the details of Performer objects from the Rock genre with payment > RM 10000.00

        System.out.println("\n\tPerformers from the Rock genre with payment > RM 10000.00:");

        for (Artist artist : artists) {

            if (artist instanceof Performer) {

                Performer performer = (Performer) artist;

                if (performer.getGenre().equalsIgnoreCase("Rock") && performer.getPayment() > 10000.00) {

                    System.out.println(performer);

                    System.out.println();

                }

            }

        }

        // Display the details of Painter objects with more than 10 paintings sold

        System.out.println("\n\tPainters with more than 10 paintings sold:");

        for (Artist artist : artists) {

            if (artist instanceof Painter) {

                Painter painter = (Painter) artist;

                if (painter.getNumOfPainting() > 10) {

                    System.out.println(painter);

                    System.out.println();

                }

            }

        }

        in.close();

        in1.close();

    } // end main

} // end class

SOURCE CODE 2.2 : Class

public class Artist {

    protected String artistProfession;

    protected String artistName;

    protected double payment;

    public Artist() {

        this.artistProfession = "";

        this.artistName = "";

        this.payment = 0;

    }

    public Artist(String artistProfession, String artistName, double payment) {

        this.artistProfession = artistProfession;

        this.artistName = artistName;

        this.payment = payment;

    }

    public void setArtistProfession() {

        this.artistProfession = artistProfession;

    }

    public String getArtistProfession() {

        return artistProfession;

    }

    public void setArtistName() {

        this.artistName = artistName;

    }

    public String getArtistName() {

        return artistName;

    }

    public void setPayment() {

        this.payment = payment;

    }

    public double getPayment() {

        return payment;

    }

    public String toString() {

        return "Artist Profession: " + artistProfession + "\nArtist Name: " + artistName + "\nPayment: " + payment;

    }

} //end class

SOURCE CODE 2.3 : Painter Class

public class Painter extends Artist{

    private int numOfPainting;

    public Painter() {

        super();

        this.numOfPainting = 0;

    }

    public Painter(String artistName, double payment, int numOfPainting) {

        this.artistProfession = "Painter";

        this.artistName = artistName;

        this.payment = payment;

        this.numOfPainting = numOfPainting;

    }

    public void setNumOfPaintings() {

        this.numOfPainting = numOfPainting;

    }

    public int getNumOfPainting() {

        return numOfPainting;

    }

    public String toString() {

        return "Artist Name: " + artistName + "\nNumber of Paintings: " + numOfPainting + "\nPayment: " + payment;

    }

    public double payArtist(double payment, int numOfPainting) {

        double totalPayment = payment \* numOfPainting;

        if (numOfPainting > 5) {

            totalPayment += totalPayment \* 0.1;

        }

        return totalPayment;

    }

}

SOURCE CODE 2.4 : Performer Class

public class Performer extends Artist{

    private int workingHours;

    private String genre;

    public Performer() {

        super();

        this.workingHours = 0;

        this.genre = "";

    }

    public Performer(String artistName, double payment, int workingHours, String genre) {

        this.artistProfession = "Performer";

        this.artistName = artistName;

        this.payment = payment;

        this.workingHours = workingHours;

        this.genre = genre;

    }

    public void setWorkingHours() {

        this.workingHours = workingHours;

    }

    public void setGenre() {

        this.genre = genre;

    }

    public int getWorkingHours() {

        return workingHours;

    }

    public String getGenre() {

        return genre;

    }

    public String toString() {

        return "Artist Name: " + artistName + "\nWorking Hours: " + workingHours + "\nPayment: " + payment;

    }

    public double payArtist(double payment, int workingHours) {

        double totalPayment = payment \* workingHours;

        if (workingHours > 24) {

            totalPayment += 500;

        }

        return totalPayment;

    }

} //end class

QUESTION 6.3

SOURCE CODE 3.1 : App Class

import java.util.Scanner;

public class App {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in); //for string

        Scanner in1 = new Scanner(System.in); //for number

        CarRental[] carRentals = new CarRental[10];

        double charge = 0;

        double sumCharge = 0;

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

            System.out.println();

            System.out.println("\tCustomer " + (i + 1));

            System.out.print("Enter Customer IC No: ");

            String custICNo = in.nextLine();

            System.out.print("Enter Customer Name: ");

            String custName = in.nextLine();

            System.out.print("Enter Customer Phone No: ");

            String custPhoneNo = in.nextLine();

            System.out.print("Enter Rental Period: ");

            double period = in1.nextDouble();

            System.out.print("Driver [true/false]: ");

            boolean driver = in1.nextBoolean();

            carRentals[i] = new CarRental(custICNo, custName, custPhoneNo, period, driver);

            charge = carRentals[i].calcCharge();

            System.out.printf("Charge: RM%.2f\n", charge);

            sumCharge += charge;

        } //end for

        System.out.println();

        System.out.printf("Total charge: RM%.2f\n", sumCharge);

        System.out.println();

        System.out.println("Customers who rented cars with the driver option:");

        for (CarRental carRental : carRentals) {

            if (carRental.getDriver()) {

                System.out.println(carRental.getCustName());

            }

        } //end for

    } //end main

} //end class

SOURCE CODE 3.2 : RentalService Class

public class RentalService {

    protected String custICNo;

    protected String custName;

    protected String custPhoneNo;

    public RentalService(String custICNo, String custName, String custPhoneNo) {

        this.custICNo = custICNo;

        this.custName = custName;

        this.custPhoneNo = custPhoneNo;

    }

    public void setCustICNo(String custICNo) {

        this.custICNo = custICNo;

    }

    public void setCustName(String custName) {

        this.custName = custName;

    }

    public void setCustPhoneNo(String custPhoneNo) {

        this.custPhoneNo = custPhoneNo;

    }

    public String getCustICNo() {

        return custICNo;

    }

    public String getCustName() {

        return custName;

    }

    public String getCustPhoneNo() {

        return custPhoneNo;

    }

    public String toString() {

        return "Customer IC No: " + custICNo + "\nCustomer Name: " + custName + "\nCustomer Phone No: " + custPhoneNo;

    }

}

SOURCE CODE 3.3 : CarRental Class

public class CarRental extends RentalService{

    private double period;

    private boolean driver;

    public CarRental(String custICNo, String custName, String custPhoneNo, double period, boolean driver) {

        super(custICNo, custName, custPhoneNo);

        this.period = period;

        this.driver = driver;

    }

    public void setPeriod(double period) {

        this.period = period;

    }

    public void setDriver(boolean driver) {

        this.driver = driver;

    }

    public double getPeriod() {

        return period;

    }

    public boolean getDriver() {

        return driver;

    }

    public String toString() {

        return "Period: " + period + "\nDriver: " + driver;

    }

    public double calcCharge() {

        double charge = 0;

        if (driver == true) {

            charge = 150;

        }

        if (period <= 6) {

            charge += 185 \* period;

        } else if (period > 6 && period <= 12) {

            charge += 230 \* period;

        } else if (period > 12 && period <= 24) {

            charge += 285 \* period;

        } else if (period > 24) {

            charge += 285 + (25 \* (period - 24));

        }

        return charge;

    }

}

QUESTION 6.4

SOURCE CODE 4.1 : Main Class

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        SoHo[] buyer = new SoHo[320];

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

            System.out.println();

            System.out.println("\tBuyer " + (i + 1));

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

            String buyerName = in.nextLine();

            System.out.print("Enter buyer IC number: ");

            long ic = in.nextLong();

            in.nextLine();

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

            String gender = in.nextLine();

            System.out.print("Bumiputera [true/false]: ");

            boolean bumiputera = in.nextBoolean();

            System.out.print("Enter unit price: ");

            double unitPrice = in.nextDouble();

            in.nextLine();

            System.out.print("Enter package [F/P/U]: ");

            char packages = in.next().charAt(0);

            in.nextLine();

            buyer[i] = new SoHo(buyerName, ic, gender, bumiputera, unitPrice, packages);

        }

        System.out.println();

        System.out.println("\tSoHo female buyers who bought fully furnished package: ");

        int cntFemale = 0;

        for (SoHo soho : buyer) {

            if (soho.getGender().equalsIgnoreCase("female")) {

                if (soho.getPackages() == 'F' || soho.getPackages() == 'f') {

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

                    cntFemale++;

                }

            }

        }

        System.out.println("Total number of SoHo female buyers who bought fully furnished package: " + cntFemale);

        in.close();

    }

}

SOURCE CODE 4.2 : House Class

public class House {

    private String buyerName;

    private long ic;

    private String gender;

    private boolean bumiputera;

    public House(String buyerName, long ic, String gender, boolean bumiputera) {

        this.buyerName = buyerName;

        this.ic = ic;

        this.gender = gender;

        this.bumiputera = bumiputera;

    }

    public String getBuyerName() {

        return buyerName;

    }

    public long getIc() {

        return ic;

    }

    public String getGender() {

        return gender;

    }

    public boolean isBumiputera() {

        return bumiputera;

    }

    public String toString() {

        return "Buyer Name: " + buyerName + "\nIC Number: " + ic + "\nGender: " + gender + "\nBumiputera: " + bumiputera + "\n";

    }

}

SOURCE CODE 4.3 : ResidentialSuites Class

public class ResidentialSuites extends House{

    private String residentialSuiteType; //A: 1 room, B: 2 rooms, C: 3 rooms

    public ResidentialSuites(String buyerName, long ic, String gender, boolean bumiputera, String residentialSuiteType) {

        super(buyerName, ic, gender, bumiputera);

        this.residentialSuiteType = residentialSuiteType;

    }

    public String getResidentialSuiteType() {

        return residentialSuiteType;

    }

    public String toString() {

        return super.toString() + "Residential Suite Type: " + residentialSuiteType + "\n";

    }

    public double residentialPrice() {

        double price = 0;

        return price;

    }

}

SOURCE CODE 4.4 : SoHo Class

public class SoHo extends House{

    private double unitPrice;

    private char packages;

    public SoHo(String buyerName, long ic, String gender, boolean bumiputera, double unitPrice, char packages) {

        super(buyerName, ic, gender, bumiputera);

        this.unitPrice = unitPrice;

        this.packages = packages;

    }

    public double getUnitPrice() {

        return unitPrice;

    }

    public char getPackages() {

        return packages;

    }

    public String toString() {

        return super.toString() + "Unit Price: " + unitPrice + "\nPackages: " + packages + "\n";

    }

    public double sohoPrice() {

        double price = 0;

        if (packages == 'F' || packages == 'f') {

            price = unitPrice + 50000.00;

        } else if (packages == 'P' || packages == 'p') {

            price = unitPrice + 20000.00;

        } else if (packages == 'U' || packages == 'u') {

            price = unitPrice;

        } else {

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

        }

        if (super.isBumiputera()) { // bumiputera buyers get 10% discount

            price = price - (price \* 0.10);

        }

        return price;

    }

}

QUESTION 6.5

SOURCE CODE 5.1 : Main Class

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in); // for strings

        Scanner in1 = new Scanner(System.in); // for numbers

        DineInCustomer[] dine = new DineInCustomer[50];

        DeliveryCustomer[] deliver = new DeliveryCustomer[50];

        System.out.println("\tEnter dine-in information");

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

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

            String name = in.nextLine();

            System.out.print("Enter package menu: ");

            char menu = in.nextLine().charAt(0);

            System.out.print("Enter number of adult: ");

            int adult = in1.nextInt();

            System.out.print("Enter number of child: ");

            int child = in1.nextInt();

            System.out.print("Enter other package: ");

            char other = in.nextLine().charAt(0);

            dine[i] = new DineInCustomer(name, menu, adult, child, other);

        }

        double sumPriceDelivery = 0;

        System.out.println("\n\tEnter delivery information");

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

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

            String name = in.nextLine();

            System.out.print("Enter package menu: ");

            char menu = in.nextLine().charAt(0);

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

            String address = in.nextLine();

            System.out.print("Enter distance (km): ");

            double km = in1.nextDouble();

            sumPriceDelivery += km;

            deliver[i] = new DeliveryCustomer(name, menu, address);

        }

        int cntDineIn = 0;

        System.out.println("\nDine-in information");

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

            if (dine[i].getCustomerName() != null) {

                System.out.println(dine[i]);

                cntDineIn++;

            }

        }

        System.out.println("Total number of dine-in customer: " + cntDineIn);

        System.out.println("Total price of all delivery customers: RM" + sumPriceDelivery);

    }

}

SOURCE CODE 5.2 : Customer Class

public class Customer {

    protected String customerName;

    private char packageMenu;

    public Customer(String customerName, char packageMenu) {

        this.customerName = customerName;

        this.packageMenu = packageMenu;

    }

    public String getCustomerName() {

        return customerName;

    }

    public char getPackageMenu() {

        return packageMenu;

    }

    public String toString() {

        return "Customer Name: " + customerName + "\nPackage Menu: " + packageMenu;

    }

}

SOURCE CODE 5.3 : DeliveryCustomer Class

public class DeliveryCustomer extends Customer{

    private String address;

    public DeliveryCustomer(String customerName, char packageMenu, String address) {

        super(customerName, packageMenu);

        this.address = address;

    }

    public String getAddress() {

        return address;

    }

    public String toString() {

        return super.toString() + "\nAddress: " + address;

    }

    public double calculatePrice(double km) {

        double price = 0;

        switch (getPackageMenu()) {

            case 'A':

            case 'a':

                price = 100;

                break;

            case 'B':

            case 'b':

                price = 150;

                break;

            case 'C':

            case 'c':

                price = 200;

                break;

            default:

                System.out.println("Invalid package menu");

                break;

        }

        if (km > 15) {

            price += 10;

        }

        return price;

    }

}

SOURCE CODE 5.4 : DineInCustomer Class

public class DineInCustomer extends Customer{

    private int numOfAdult;

    private int numOfChild;

    private char otherPackage;

    public DineInCustomer(String customerName, char packageMenu, int numOfAdult, int numOfChild, char otherPackage) {

        super(customerName, packageMenu);

        this.numOfAdult = numOfAdult;

        this.numOfChild = numOfChild;

        this.otherPackage = otherPackage;

    }

    public int getNumOfAdult() {

        return numOfAdult;

    }

    public int getNumOfChild() {

        return numOfChild;

    }

    public char getOtherPackage() {

        return otherPackage;

    }

    public String toString() {

        return super.toString() + "\nNumber of Adult: " + numOfAdult + "\nNumber of Child: " + numOfChild + "\nOther Package: " + otherPackage;

    }

}

QUESTION 6.6

SOURCE CODE 6.1 : Main Class

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in); // for strings

        Scanner in1 = new Scanner(System.in); // for numbers

        RoomActivity[] room = new RoomActivity[2];

        System.out.println("\tEnter room activity information");

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

            System.out.println();

            System.out.println("\tRoom " + (i + 1));

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

            String activityName = in.nextLine();

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

            char organiser = in.nextLine().charAt(0);

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

            double budget = in1.nextDouble();

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

            String date = in.nextLine();

            System.out.print("Enter room type: ");

            String roomType = in.nextLine();

            System.out.print("Enter number of participants: ");

            int numOfParticipants = in1.nextInt();

            room[i] = new RoomActivity(activityName, organiser, budget, date, roomType, numOfParticipants);

        }

        System.out.println();

        System.out.println("\tBooked Activities");

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

            if (room[i] != null) {

            System.out.println("Activity Name: " + room[i].getActivityName());

            }

        }

        System.out.println();

        System.out.println("\tBookings by Private Organisers");

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

            if (room[i] != null && room[i].getOrganiser() == 'P' || room[i].getOrganiser() == 'p') {

                System.out.println(room[i].toString());

                System.out.println("Charges: " + room[i].roomCharges());

                System.out.println();

            }

        }

        System.out.println("\tBookings for lab with more than 30 participants");

        int cntLab = 0;

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

            if (room[i] != null && room[i].getRoomType().equalsIgnoreCase("lab") && room[i].getNumOfParticipants() > 30) {

                System.out.println(room[i].toString());

                System.out.println();

                cntLab++;

            }

        }

        System.out.println("Total number of lab with more than 30 participants: " + cntLab);

        in.close();

        in1.close();

    }

}

SOURCE CODE 6.2 : RoomActivity Class

public class RoomActivity extends Activity{

    private String roomType;

    private int numOfParticipants;

    public RoomActivity(String activityName, char organiser, double budget, String date, String roomType, int numOfParticipants) {

        super(activityName, organiser, budget, date);

        this.roomType = roomType;

        this.numOfParticipants = numOfParticipants;

    }

    public String getRoomType() {

        return roomType;

    }

    public int getNumOfParticipants() {

        return numOfParticipants;

    }

    public String toString() {

        return super.toString() + "\nRoom Type: " + roomType + "\nNumber of Participants: " + numOfParticipants;

    }

    public double roomCharges() {

        double charge = 0;

        switch (getOrganiser()) {

            case 'G':

            case 'g':

                charge = 1500;

                break;

            case 'P':

            case 'p':

                charge = 25 \* numOfParticipants;

                break;

            case 'U':

            case 'u':

                charge = 10 \* numOfParticipants;

                break;

            default:

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

                break;

        }

        return charge;

    }

}

SOURCE CODE 6.3 : Activity Class

public class Activity {

    private String activityName;

    private char organiser;

    private double budget;

    private String date;

    public Activity(String activity, char organiser, double budget, String date) {

        this.activityName = activityName;

        this.organiser = organiser;

        this.budget = budget;

        this.date = date;

    }

    public String getActivityName() {

        return activityName;

    }

    public char getOrganiser() {

        return organiser;

    }

    public double getBudget() {

        return budget;

    }

    public String getDate() {

        return date;

    }

    public String toString() {

        return "Activity: " + activityName + "\nOrganiser: " + organiser + "\nBudget: " + budget + "\nDate: " + date;

    }

}

QUESTION 6.6

SOURCE CODE 7.1 : Main Class

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner intInput = new Scanner(System.in);

        Scanner strInput = new Scanner(System.in);

        onlineClass[] onlineClass = new onlineClass[100];

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

            onlineClass[i] = new onlineClass("", "", 0, 'A', 0, "");

        }

        System.out.println("Welcome to the course registration system");

        System.out.print("\nEnter amount of students: ");

        int amountOfStudents = intInput.nextInt();

        for (int i = 0; i < amountOfStudents; i++) {

            System.out.print("Enter student ID: ");

            String studentID = strInput.nextLine();

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

            String studentName = strInput.nextLine();

            System.out.print("Enter student age: ");

            int studentAge = intInput.nextInt();

            System.out.print("Enter student gender (M/F): ");

            char studentGender = strInput.nextLine().charAt(0);

            System.out.print("\n1. Undergraduate\n2. Postgraduate\n\nEnter student type (1-2): ");

            int studentType = intInput.nextInt();

            System.out.print("\n1. COMP\n2. ENG\n\nEnter program code (1-2): ");

            int programCodeInt = intInput.nextInt();

            String programCode = "";

            switch (programCodeInt) {

                case 1:

                    programCode = "COMP";

                    break;

                case 2:

                    programCode = "ENG";

                    break;

                default:

                    break;

            }

            onlineClass[i] = new onlineClass(studentID, studentName, studentAge, studentGender, studentType,

                    programCode);

            System.out.println();

        }

        for (int i = 0; i < amountOfStudents; i++) {

            if (onlineClass[i].getStudentID().equals("2017111222")) {

                onlineClass[i].setProgramCode("COMP");

                break;

            }

        }

        double totalFee = 0;

        for (int i = 0; i < amountOfStudents; i++) {

            switch (onlineClass[i].getProgramCode()) {

                case "ENG":

                    if (onlineClass[i].getStudentType() == 1) {

                        totalFee += onlineClass[i].calculateFee();

                    }

                    break;

                default:

                    break;

            }

        }

        System.out.printf("Total fee collected from engineering program for undergraduate students: RM %.2f\n",

                totalFee);

        for (int i = 0; i < amountOfStudents; i++) {

            if (onlineClass[i].getStudentName().equals("Alexander")) {

                System.out.println(

                        "Program code enrolled by a student named 'Alexander': " + onlineClass[i].getProgramCode());

                break;

            }

        }

        intInput.close();

        strInput.close();

    }

}

SOURCE CODE 7.2 : courseRegistration Class

public abstract class courseRegistration {

    private String studentID;

    private String studentName;

    private int studentAge;

    private char studentGender;

    private int studentType;

    public courseRegistration(String studentID, String studentName, int studentAge, char studentGender,

            int studentType) {

        this.studentID = studentID;

        this.studentName = studentName;

        this.studentAge = studentAge;

        this.studentGender = studentGender;

        this.studentType = studentType;

    }

    public String getStudentID() {

        return this.studentID;

    }

    public void setStudentID(String studentID) {

        this.studentID = studentID;

    }

    public String getStudentName() {

        return this.studentName;

    }

    public void setStudentName(String studentName) {

        this.studentName = studentName;

    }

    public int getStudentAge() {

        return this.studentAge;

    }

    public void setStudentAge(int studentAge) {

        this.studentAge = studentAge;

    }

    public char getStudentGender() {

        return this.studentGender;

    }

    public void setStudentGender(char studentGender) {

        this.studentGender = studentGender;

    }

    public int getStudentType() {

        return this.studentType;

    }

    public void setStudentType(int studentType) {

        this.studentType = studentType;

    }

    public abstract double calculateFee();

    public String toString() {

        return "Student ID: " + studentID + "\nStudent Name: " + studentName + "\nStudent Age: " + studentAge

                + "\nGender: " + studentGender + "\nStudent Type: " + studentType;

    }

}

SOURCE CODE 7.3 : onlineClass Class

public class onlineClass extends courseRegistration {

    private String programCode;

    public onlineClass(String studentID, String studentName, int studentAge, char studentGender, int studentType,

            String programCode) {

        super(studentID, studentName, studentAge, studentGender, studentType);

        this.programCode = programCode;

    }

    public String getProgramCode() {

        return this.programCode;

    }

    public void setProgramCode(String programCode) {

        this.programCode = programCode;

    }

    public double calculateFee() {

        double fee = 0;

        if (getStudentType() == 1) {

            fee = 1600;

            if (programCode.equals("COMP")) {

                fee \*= 0.6;

            } else if (programCode.equals("ENG")) {

                fee \*= 0.35;

            }

        } else if (getStudentType() == 2) {

            fee = 2100;

            if (programCode.equals("COMP")) {

                fee \*= 0.7;

            } else if (programCode.equals("ENG")) {

                fee \*= 0.75;

            }

        }

        return fee;

    }

    public String toString() {

        return super.toString() + "\nProgram Code: " + programCode;

    }

}