**Problem 1 : Accelerate the Car (20 Marks) a)**

class Car{

private int year;

private String make;

private double speed;

public Car(int year,String make,double beginning\_speed){

this.year=year;

this.make=make;

this.speed=beginning\_speed;

}

public int getYear(){

return this.year;

}

public String getMake(){

return this.make;

}

public double getSpeed(){

return this.speed;

}

public void accelerate(){

this.speed++;

}

}

class RaceTrack{

public static void main(String args[]){

Car c=new Car(2010,"Porsche",25.0);

System.out.println("Car's Year Model"+" "+c.getYear());

System.out.println("Make of the Car"+" "+c.getMake());

System.out.println("Car's Current Speed"+" "+c.getSpeed());

c.accelerate();

System.out.println("Modified Speed"+" "+c.getSpeed());

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***OUTPUT**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

D:\CDAC\JAVA\_EXAM>javac RaceTrack.java

D:\CDAC\JAVA\_EXAM>java RaceTrack

Car's Year Model 2010

Make of the Car Porsche

Car's Current Speed 25.0

Modified Speed 26.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Problem 2 : Inventory Management (20 Marks)**

**Write a program to create an inventory of items which will allow basic inventory management.**

import java.util.\*;

class Item{

Integer item\_id;

String item\_name;

Item(int itemId,String itemName){

this.item\_id=itemId;

this.item\_name=itemName;

}

Item(){

}

void setItemId(int id){

this.item\_id=id;

}

void setItemName(String name){

this.item\_name=name;

}

@Override

public String toString(){

return this.item\_id+" "+this.item\_name;

}

@Override

public boolean equals(Object o){

if(o instanceof Item){

Item temp=(Item)o;

if(this.item\_id.equals(temp.item\_id) && this.item\_name.equals(temp.item\_name)){

return true;

}

}

return false;

}

@Override

public int hashCode(){

int prime=13;

int val=1;

val=val\*prime + this.item\_id.hashCode();

val=val\*prime + this.item\_name.hashCode();

return val;

}

}

class ItemID\_sort implements Comparator<Item>{

public int compare(Item I1,Item I2){

return I1.item\_id-(I2.item\_id);

}

}

class ItemName\_sort implements Comparator<Item>{

public int compare(Item I1,Item I2){

return I1.item\_name.compareTo(I2.item\_name);

}

}

import java.util.\*;

class Inventory{

public static void main(String args[]){

Item I=new Item();

Scanner sc=new Scanner(System.in);

int ch;

Item i1=new Item(1,"abc");

Item i2=new Item(4,"pqr");

Item i3=new Item(3,"xyz");

Item i4=new Item(2,"cpr");

ArrayList<Item> list= new ArrayList<>();

list.add(i1);

list.add(i2);

list.add(i3);

list.add(i4);

System.out.println("Arraylist"+" "+list);

do{

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

System.out.println("1.Add Item\n2.Display complete inventory in sorted order of item names as well as itemId\n3.Remove Item\n4.Exit");

ch=sc.nextInt();

switch(ch){

case 1: System.out.println("Enter ItemId");

int id=sc.nextInt();

I.setItemId(id);

System.out.println("Enter ItemName");

sc.nextLine();

String name=sc.nextLine();

I.setItemName(name);

if(!list.contains(I)){

list.add(I);

}

System.out.println("Arraylist"+" "+list);

break;

case 2: System.out.println("Sorting by ItemID");

ItemID\_sort idsort=new ItemID\_sort();

Collections.sort(list,idsort);

System.out.println(list);

System.out.println("Sorting by ItemName");

ItemName\_sort iname=new ItemName\_sort();

Collections.sort(list,iname);

System.out.println(list);

break;

case 3: System.out.println("Enter index to remove Item");

int i=sc.nextInt();

list.remove(i);

System.out.println("Updated ArrayList");

System.out.println(list);

break;

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

break;

}

}while(ch!=4);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***OUTPUT**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

D:\CDAC\JAVA\_EXAM>javac Inventory.java

D:\CDAC\JAVA\_EXAM>java Inventory

Arraylist [1 abc, 4 pqr, 3 xyz, 2 cpr]

Enter your choice

1.Add Item

2.Display complete inventory in sorted order of item names as well as itemId

3.Remove Item

4.Exit

1

Enter ItemId

3

Enter ItemName

xyz

Arraylist [1 abc, 4 pqr, 3 xyz, 2 cpr]

Enter your choice

1.Add Item

2.Display complete inventory in sorted order of item names as well as itemId

3.Remove Item

4.Exit

2

Sorting by ItemID

[1 abc, 2 cpr, 3 xyz, 4 pqr]

Sorting by ItemName

[1 abc, 2 cpr, 4 pqr, 3 xyz]

Enter your choice

1.Add Item

2.Display complete inventory in sorted order of item names as well as itemId

3.Remove Item

4.Exit

3

Enter index to remove Item

2

Updated ArrayList

[1 abc, 2 cpr, 3 xyz]

Enter your choice

1.Add Item

2.Display complete inventory in sorted order of item names as well as itemId

3.Remove Item

4.Exit

4

Exit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*