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

**Class Car**

class Car{

private int year;

private String make;

private double speed;

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

this.year=year;

this.make=make;

this.speed=beg\_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 race track**

class RaceTrack{

public static void main(String... args){

Car c1=new Car(2010, "Porche",25.0);

System.out.println("Manufacturing year :"+ c1.getYear());

System.out.println("Car maker :" +c1.getMake());

System.out.println("Car Speed :" +c1.getSpeed());

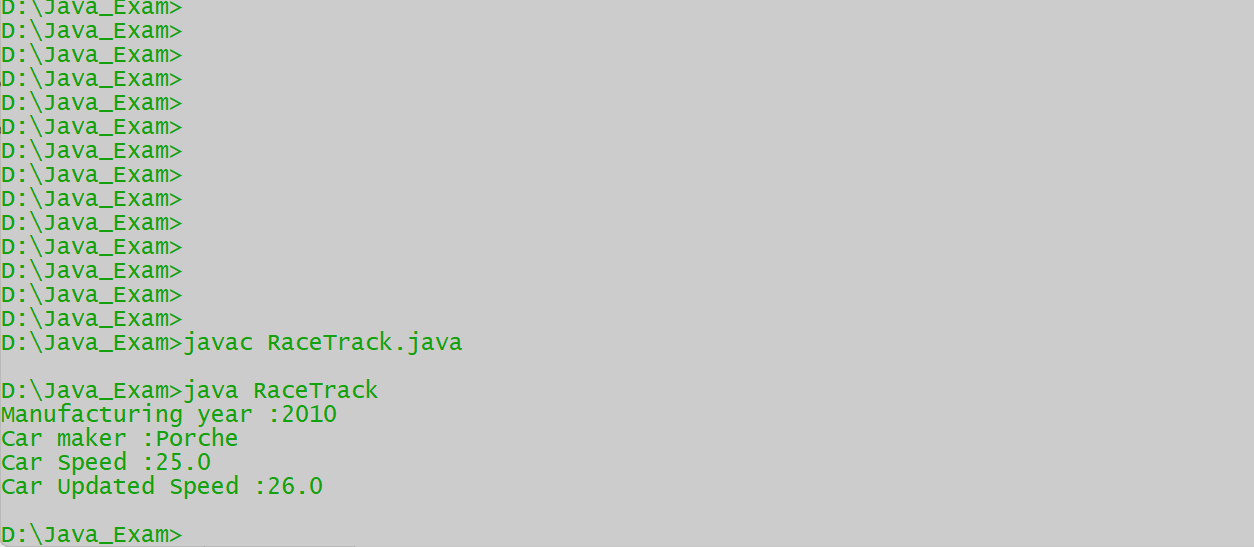
c1.Accelerate();

System.out.println("Car Updated Speed :" +c1.getSpeed());

}

}

Output



import java.util.\*;

class Item{

Integer item\_id;

String item\_name;

Item(int item\_id, String item\_name){

this.item\_id=item\_id;

this.item\_name=item\_name;

}

Item(){}

void setitem\_id(int item\_id){

this.item\_id=item\_id;

}

void setitem\_name(String item\_name){

this.item\_name=item\_name;

}

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;

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

}

@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 namesort implements Comparator<Item>{

public int compare(Item I1, Item I2){

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

}

}

class idsort implements Comparator<Item>{

public int compare(Item I1, Item I2){

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

}

}

class Inventory{

static Item I=new Item();

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

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int choice;

Item I1= new Item(1,"ItemA");

Item I2= new Item(2,"ItemB");

Item I4= new Item(4,"ItemD");

Item I3= new Item(3,"ItemC");

list.add(I1);

list.add(I2);

list.add(I4);

list.add(I3);

System.out.println(list);

do{

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

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

choice= sc.nextInt();

switch(choice){

case 1:

System.out.println("Enter the details as follows");

System.out.println("Enter Item you want");

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

System.out.println("Please enter the Itemid ");

int d=sc.nextInt();

I.setitem\_id(d);

System.out.println("Please enter the Itemname ");

sc.nextLine();

String itn=sc.nextLine();

I.setitem\_name(itn);

if(!list.contains(I)){

list.add(I);}

}

System.out.println("Item added are as bellow ");

System.out.println( list);

break;

case 2:

System.out.println(" List before sorting");

System.out.println(list);

System.out.println("List sorted by Itemid");

idsort m2=new idsort();

Collections.sort(list,m2);

System.out.println(list);

System.out.println("List sorted by Itemname");

namesort m1=new namesort();

Collections.sort(list,m1);

System.out.println(list);

break;

case 3:

System.out.println("Current list");

System.out.println(list);

System.out.println("To remove Item Please enter the index of item");

int rem=sc.nextInt();

list.remove(rem);

System.out.println("List after removing the Item");

System.out.println(list);

break;

case 4:

System.out.println("Exit");

break;

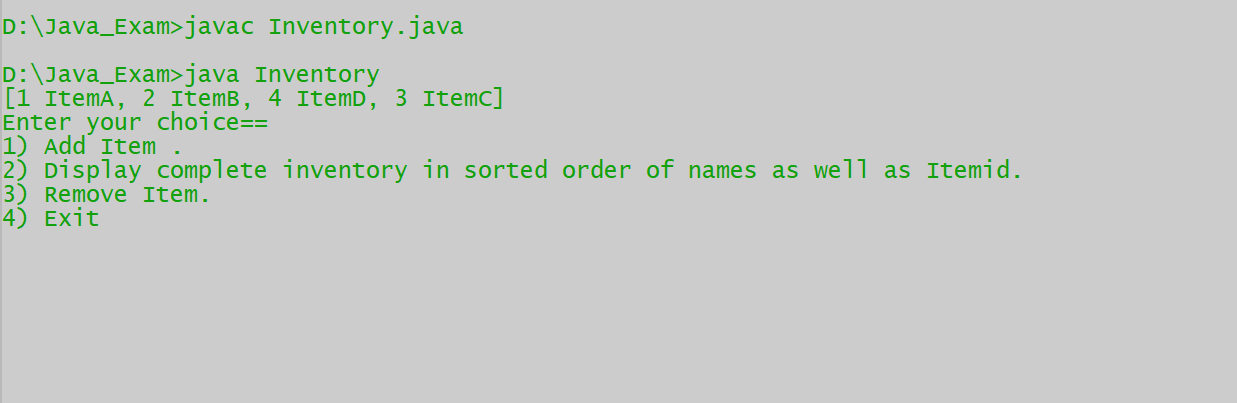
}

}while(choice!=4);

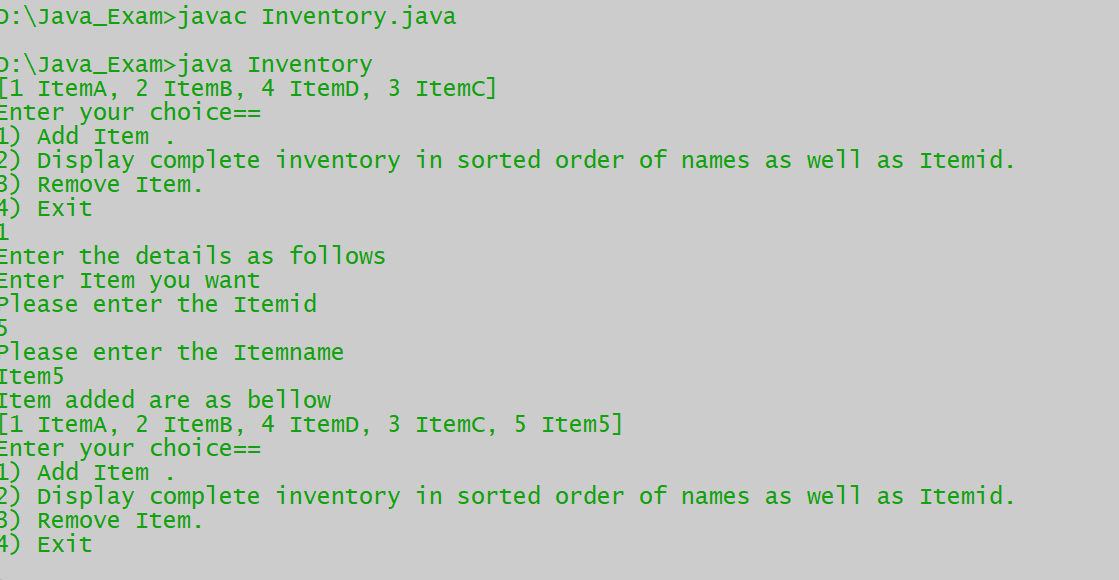
}

}

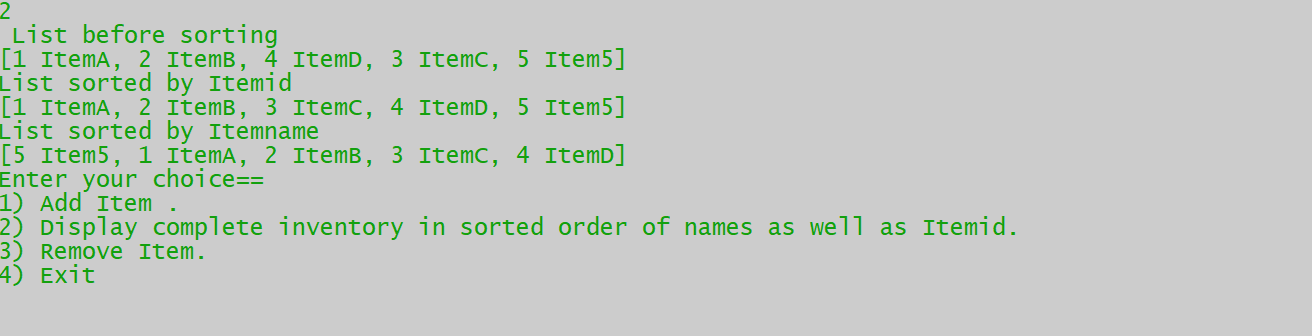
Output



Adding Item



Displaying sorted list



Removing items through index

