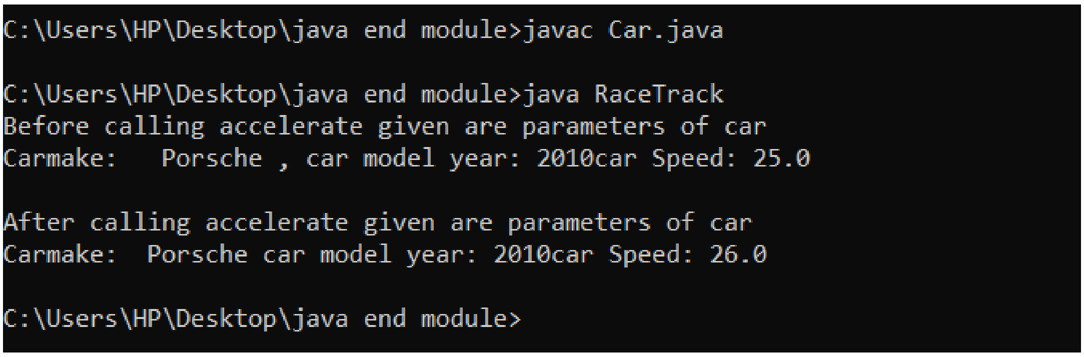
Q.1

```
class Car {
private int year;
private String make;
private double speed;

Car(int year, String make, double speed)
{
this.year=year;
this.make=make;
this.speed=speed;
}

int getYear(){
return this.year;
}
void setYear(int year)
{
this.year=year;
}



String getMake(){
return this.make;
}
void setMake(String make)
{
this.make=make;
}
```

```
double getSpeed(){
return this.speed;
}
void setYear(double speed)
{
this.speed=speed;
}



void accelerate(){
        this.speed=this.speed+1;
}


}




class RaceTrack{


        public static void main (String [] args)
        {


                System.out.println("Before calling accelerate given are parameters of car");
                Car c = new Car(2010,"Porsche",25.0);
                System.out.println("Carmake:   "+c.getMake()+" , "+"car model year: "+c.getYear()+
"car Speed: "+c.getSpeed());


                System.out.println("   ");
                c.accelerate();
                System.out.println("After calling accelerate given are parameters of car");
                System.out.println("Carmake:  "+c.getMake()+" "+"car model year: "+c.getYear()+
"car Speed: "+c.getSpeed());
```

```
        }
}
```



```
C:\Users\HP\Desktop\java end module>javac Car.java

C:\Users\HP\Desktop\java end module>java RaceTrack
Before calling accelerate given are parameters of car
Carmake:   Porsche , car model year: 2010car Speed: 25.0

After calling accelerate given are parameters of car
Carmake:  Porsche car model year: 2010car Speed: 26.0

C:\Users\HP\Desktop\java end module>
```

Q.2

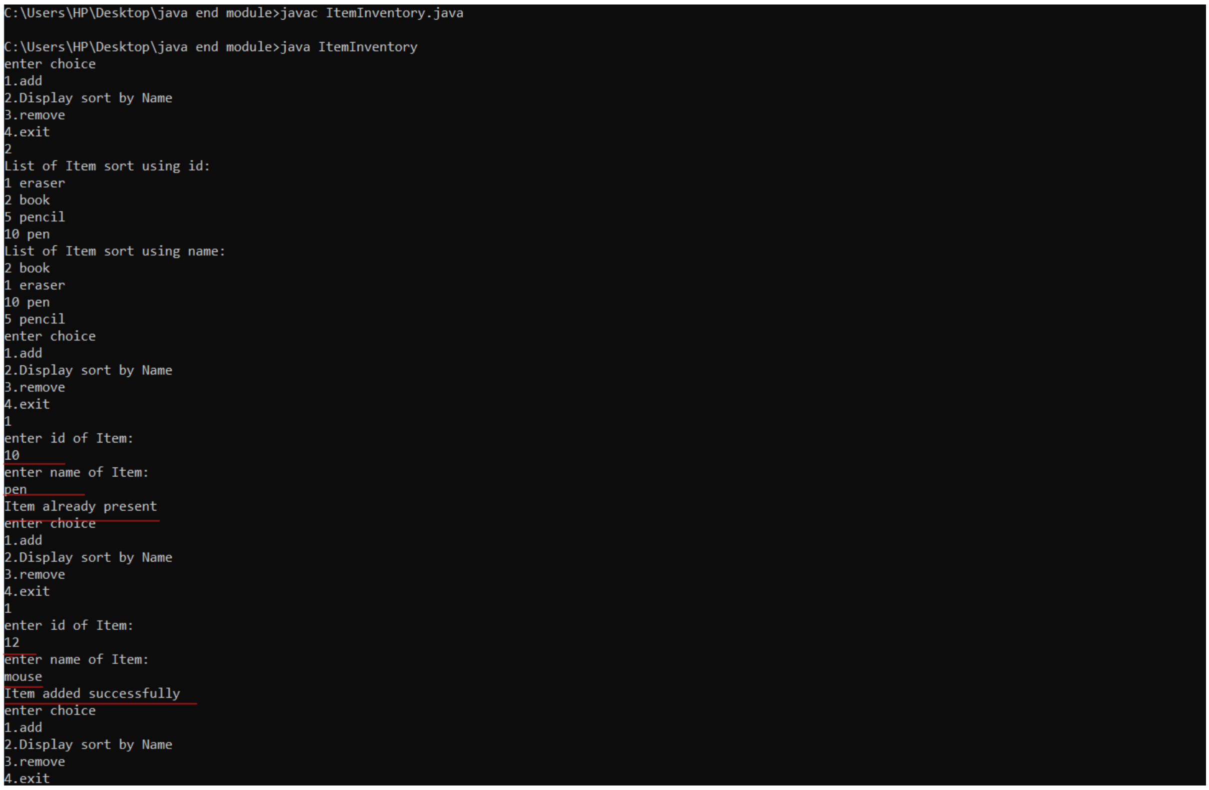
```
import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Scanner;


class Item

{

        int itemId;

        String itemName;


        Item(int itemId,String itemName)

        {

                this.itemId=itemId;

                this.itemName=itemName;


        }
```

```
public int getItemId()
{
        return this.itemId;
}



public void setItemId(int itemId)
{
        this.itemId=itemId;
}

@Override
public int hashCode()
{
        final int prime=31;
        int result=12;
        result=prime*result+ itemId;
        result=prime*result+ ((itemName==null)? 0:itemName.hashCode());
        return result;

}

@Override
public boolean equals (Object obj)
{
        Item temp=(Item) obj;

        if ((this.itemId==temp.itemId) && ((this.itemName.equals(temp.itemName))))
        {
        return true;
```

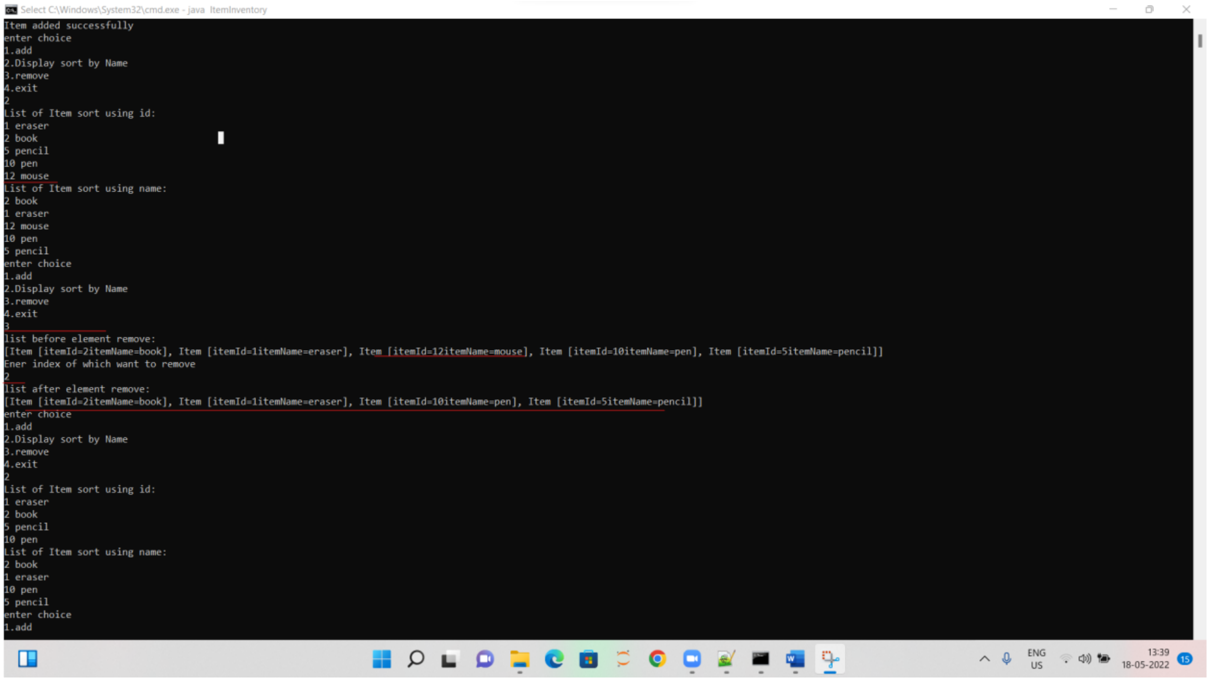
```
		}
		return false;
	}

	@Override
	public String toString()
	{
		return "Item [itemId=" +itemId+"itemName=" +itemName+"]" ;
	}
	public String getItemName()
	{
		return itemName;
	}



	public void setItemName(String itemName)
	{
		this.itemName=itemName;
	}
}



class SortById implements Comparator<Item>
{

	@Override
	public int compare(Item o1,Item o2)
	{
		return o1.itemId-(o2.itemId);
	}
}
```

```
class SortByName implements Comparator<Item>
{
        @Override
        public int compare(Item o1, Item o2)
        {
        return o1.itemName.compareTo(o2.itemName);
        }
}

public class ItemInventory
{

        public static void main(String [] args)
        {
                ArrayList<Item> itemList=new ArrayList<>();
                itemList.add(new Item(10,"pen"));
                itemList.add(new Item(2,"book"));
                itemList.add(new Item(5,"pencil"));
                itemList.add(new Item(1,"eraser"));

                Scanner sc=new Scanner(System.in);
                int choice;
        do{
                System.out.println("enter choice\n1.add\n2.Display sort by
Name\n3.remove\n4.exit ");
                choice=sc.nextInt();
                switch(choice)
                {
                case 1:
                                System.out.println("enter id of Item:");
                                int id =sc.nextInt();
                                System.out.println("enter name of Item:");
                                String name =sc.next();

                                Item i= new Item(id,name);

                                if (!itemList.contains(i))
                                {
                                    itemList.add(i);
                                }
                                System.out.println("Item added successfully");
                                break;
                        case 2:
                                System.out.println("List of Item sort using id:");
                                Collections.sort(itemList,new SortById());

                                for (Item item: itemList)
                                {
                                    System.out.println(item.getItemId()+ "
"+item.getItemName());
                                }

                                System.out.println("List of Item sort using name:");
                                Collections.sort(itemList,new SortByName());

                                for (Item item: itemList){
                                    System.out.println(item.getItemId()+ "
"+item.getItemName());
                                }
                                break;
                case 3:
                                System.out.println("list before element remove:");
```

```
                        System.out.println(itemList);
                        System.out.println("Ener index of which want to remove");
                         int index= sc.nextInt();
                         itemList.remove(index);
                         System.out.println("list after element remove:");
                        System.out.println(itemList);
                        break;


                case 4:
                        System.out.println("Thank you!");
                        break;
                default:
                        break;


                }
            } while (choice!=4);

    }
    }
```



```
C:\Users\HP\Desktop\java end module>javac ItemInventory.java

C:\Users\HP\Desktop\java end module>java ItemInventory
enter choice
1.add
2.Display sort by Name
3.remove
4.exit
2
List of Item sort using id:
1 eraser
2 book
5 pencil
10 pen
List of Item sort using name:
2 book
1 eraser
10 pen
5 pencil
enter choice
1.add
2.Display sort by Name
3.remove
4.exit
1
enter id of Item:
10
enter name of Item:
pen
Item already present
enter choice
1.add
2.Display sort by Name
3.remove
4.exit
1
enter id of Item:
12
enter name of Item:
mouse
Item added successfully
enter choice
1.add
2.Display sort by Name
3.remove
4.exit
```



```
Item added successfully
enter choice
1.add
2.Display sort by Name
3.remove
4.exit
2
List of Item sort using id:
1 eraser
2 book
5 pencil
10 pen
12 mouse
List of Item sort using name:
2 book
1 eraser
12 mouse
10 pen
5 pencil
enter choice
1.add
2.Display sort by Name
3.remove
4.exit
3
list before element remove:
[Item [itemId=2itemName=book], Item [itemId=1itemName=eraser], Item [itemId=12itemName=mouse], Item [itemId=10itemName=pen], Item [itemId=5itemName=pencil]]
Ener index of which want to remove
2
list after element remove:
[Item [itemId=2itemName=book], Item [itemId=1itemName=eraser], Item [itemId=10itemName=pen], Item [itemId=5itemName=pencil]]
enter choice
1.add
2.Display sort by Name
3.remove
4.exit
2
List of Item sort using id:
1 eraser
2 book
5 pencil
10 pen
List of Item sort using name:
2 book
1 eraser
10 pen
5 pencil
enter choice
1.add
```