**WHOLE PRACTICE ASSIGNMENT**

**QUESTION1:**

**Bank Class:**

**package** corejava;

**import** java.util.Arrays;

**public** **class** Batch {

**public** String batchCode;

**public** String startDate;

**public** String endDate;

**public** Trainee[] trainees;

**public** Trainee[] getTrainees(String traineeID) {

Trainee[] foundTrainees=**new** Trainee[3];

**int** count=0;

**for**(Trainee trainee:trainees) {

String id=trainee.getTraineeID();

**if**(id.equals(traineeID)) {

foundTrainees[count]=trainee;

count++;

}

}

**return** foundTrainees;

}

**public** Trainee[] getTrainees(String gender) {

Trainee[] foundTrainees=**new** Trainee[3];

**int** count=0;

**for**(Trainee trainee:trainees) {

String gen=trainee.getGender();

**if**(gen.equals(gender)) {

foundTrainees[count]=trainee;

count++;

}

}

**return** foundTrainees;

}

**public** Trainee[] getTrainees(**int** age){

Trainee[] foundTrainees=**new** Trainee[3];

**int** count=0;

**for**(Trainee trainee:trainees) {

**int** traineeAge=trainee.getAge();

**if**(traineeAge==age) {

foundTrainees[count]=trainee;

count++;

}

}

**return** foundTrainees;

}

**public** String getBatchCode() {

**return** batchCode;

}

**public** **void** setBatchCode(String batchCode) {

**this**.batchCode = batchCode;

}

**public** String getStartDate() {

**return** startDate;

}

**public** **void** setStartDate(String startDate) {

**this**.startDate = startDate;

}

**public** String getEndDate() {

**return** endDate;

}

**public** **void** setEndDate(String endDate) {

**this**.endDate = endDate;

}

**public** Trainee[] getTrainees() {

**return** trainees;

}

**public** **void** setTrainees(Trainee[] trainees) {

**this**.trainees = trainees;

}

**public** Batch(String batchCode, String startDate, String endDate, Trainee[] trainees) {

**super**();

**this**.batchCode = batchCode;

**this**.startDate = startDate;

**this**.endDate = endDate;

**this**.trainees = trainees;

}

**public** Batch() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Batch [batchCode=" + batchCode + ", startDate=" + startDate + ", endDate=" + endDate + ", trainees="

+ Arrays.*toString*(trainees) + "]";

}

}

**Trainee Class:**

**package** corejava;

**public** **class** Trainee {

String traineeID;

String traineeName;

**public** **int** contactNo;

String email;

String gender;

**int** age;

**public** String getTraineeID() {

**return** traineeID;

}

**public** **void** setTraineeID(String traineeID) {

**this**.traineeID = traineeID;

}

**public** String getTraineeName() {

**return** traineeName;

}

**public** **void** setTraineeName(String traineeName) {

**this**.traineeName = traineeName;

}

**public** **int** getContactNo() {

**return** contactNo;

}

**public** **void** setContactNo(**int** contactNo) {

**this**.contactNo = contactNo;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** Trainee(String traineeID, String traineeName, **int** contactNo, String email, String gender, **int** age) {

**super**();

**this**.traineeID = traineeID;

**this**.traineeName = traineeName;

**this**.contactNo = contactNo;

**this**.email = email;

**this**.gender = gender;

**this**.age = age;

}

**public** Trainee() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Trainee [traineeID=" + traineeID + ", traineeName=" + traineeName + ", contactNo=" + contactNo + ", email="

+ email + ", gender=" + gender + ", age=" + age + "]";

}

}

**BankMain:**

package corejava;

import java.util.Arrays;

import corejava.Trainee;

public class BatchMain {

public static void main(String[] args) {

Trainee[] trainees=new Trainee[3];

trainees[0]=new Trainee("RA123","HaRi",98769,"9841501361hry@gmail.com","Male",23);

trainees[1]=new Trainee("RA124","Mani",88769,"8841501361hry@gmail.com","Male",33);

trainees[2]=new Trainee("RA125","HaRiNi",78769,"7841501361hry@gmail.com","FeMale",43);

Batch batch=new Batch("TA767","27 may","28aug",trainees);

System.out.println(Arrays.toString(batch.getTrainees(23)));

System.out.println(Arrays.toString(batch.getTrainees("Male")));

System.out.println(Arrays.toString(batch.getTrainees("RA124")));

}

}

**QUESTION2:**

**File1:Project.java---**

**public** **class** Project {

String projectID;

String projectName;

String ProjectHead;

**int** noOfResources;

**public** String getProjectID() {

**return** projectID;

}

**public** **void** setProjectID(String projectID) {

**this**.projectID = projectID;

}

**public** String getProjectName() {

**return** projectName;

}

**public** **void** setProjectName(String projectName) {

**this**.projectName = projectName;

}

**public** String getProjectHead() {

**return** ProjectHead;

}

**public** **void** setProjectHead(String projectHead) {

ProjectHead = projectHead;

}

**public** **int** getNoOfResources() {

**return** noOfResources;

}

**public** **void** setNoOfResources(**int** noOfResources) {

**this**.noOfResources = noOfResources;

}

**public** Project(String projectID,String projectName,String projectHead,**int** noOfResources) {

**super**();

**this**.projectID=projectID;

**this**.projectName=projectName;

**this**.ProjectHead=projectHead;

**this**.noOfResources=noOfResources;

}

}

**File2:ProjectMain.java---**

**public** **class** ProjectMain{

**public** **static** **void** main(String[] args) {

Project myProject=**new** Project("RA1731241020051","Tracking System","Hari K",10);

System.***out***.println(myProject.getProjectID());

System.***out***.println(myProject.getProjectName());

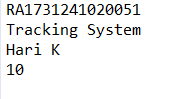
System.***out***.println(myProject.getProjectHead());

System.***out***.println(myProject.getNoOfResources());

}

}

**Output---**

****

**QUESTION 3:**

**package** corejava;

**public** **class** StringServiceProvider {

**public** **static** **void** main(String[] args) {

String name="HarriK";

**char** key='i';

**int** k;

**int** i=0;

**int** len=0;

**char** searchElement='K';

**char** replaceElement='I';

**try** {

**while**(**true**) {

**if**(name.charAt(i)!='\0')

len++;

i++;

}

}

**catch**(Exception e)

{

System.***out***.println("Length of the String:"+len);

}

System.***out***.println("Reversed String:");

**for**(**int** j=len-1;j>=0;j--) {

System.***out***.print(name.charAt(j));

}

System.***out***.println("");

**int** count=0;

**for**(k=0;k<len-1;k++) {

**if**(name.charAt(k)==key)

{

System.***out***.println("Searched Element:");

System.***out***.println("The searching word "+key+" found at the index of "+k);

count++;

}

}

**if**(count==0) {

System.***out***.println("The given word is not found");

}

StringServiceProvider s= **new** StringServiceProvider();

String replaced = s.replaced(name,searchElement,replaceElement);

System.***out***.println("After Replaced : "+replaced);

}

**public** String replaced(String name,**char** searchElement,**char** replaceElement) {

**char**[] arr=name.toCharArray();

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

**if**(arr[i]==searchElement)

{

arr[i]=replaceElement;

}

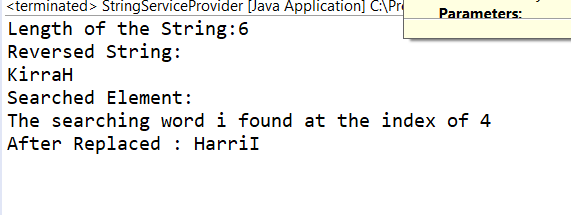
}

String replaced=**new** String(arr);

**return** replaced;

}

}



**QUESTION 4,5,6:**

**BANKACCOUNT:**

**package** overloading;

**public** **class** BankAccount {

**private** **static** **int** *lastAssignedNo*;

**static** {

*lastAssignedNo* = 100;

}

**private** String accountNo;

**public** String accountName;

**private** **double** balance = 0;

**public** String getAccountNo() {

**return** accountNo;

}

**public** **void** setAccountNo(String accountNo) {

**this**.accountNo = "" + *lastAssignedNo*++;

}

**public** String getAccountName() {

**return** accountName;

}

**public** **void** setAccountName(String accountName) {

**this**.accountName = accountName;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** BankAccount(String accountNo, String accountName, **double** balance) {

**super**();

**this**.accountNo = accountNo;

**this**.accountName = accountName;

**this**.balance = balance;

}

**public** BankAccount(String accountNo, String accountName) {

**super**();

**this**.accountNo = accountNo;

**this**.accountName = accountName;

**this**.balance = 1000;

}

**public** BankAccount() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "BankAccount [accountNo=" + accountNo + ", accountName=" + accountName + ", balance=" + balance + "]";

}

}

**BANK:**

package overloading;

import java.util.Arrays;

import overloading.InvalidAmountException;

import overloading.InvalidFundException;

import overloading.IBankServiceProvider;

public class Bank implements IBankServiceProvider {

public BankAccount[] accounts;

private String IFSC;

private String bankName;

public Bank(BankAccount[] accounts,String ifsc,String bankName) {

super();

this.accounts=accounts;

IFSC=ifsc;

this.bankName=bankName;

}

public Bank() {

super();

}

public BankAccount CheckAccount(String accountNo)throws InvalidAmountException {

BankAccount foundAccount=null;

for(BankAccount account:accounts)

{

if(account.getAccountNo().equals(accountNo)) {

foundAccount=account;

break;

}

}

if(foundAccount==null) {

throw new InvalidAmountException();

}

return foundAccount;

}

public double getBalance(String accountNo)throws InvalidAmountException {

double balance=0.0;

BankAccount foundAccount=CheckAccount(accountNo);

balance=foundAccount.getBalance();

return balance;

}

public boolean depositMoney(String accountNo,double amount)throws InvalidAmountException{

boolean depositFlag=false;

BankAccount foundAccount=CheckAccount(accountNo);

if(foundAccount!=null)

{

foundAccount.setBalance(foundAccount.getBalance()+amount);;

depositFlag=true;

}

return true;

}

public boolean withdrawMoney(String accountNo,double amount)throws InvalidAmountException,InvalidFundException{

boolean withdrawFlag=false;

BankAccount foundAccount=CheckAccount(accountNo);

if(foundAccount!=null)

{

if(foundAccount.getBalance()>amount) {

foundAccount.setBalance(foundAccount.getBalance()-amount);;

withdrawFlag=true;

}

else {

throw new InvalidFundException();

}

}

return withdrawFlag;

}

public boolean transferMoney(String fromAccountNo,String toAccountNo,double amount)throws InvalidAmountException,InvalidFundException{

boolean transferFlag=false;

boolean withdrawFlag=withdrawMoney(fromAccountNo,amount);

boolean depositFlag=depositMoney(toAccountNo,amount);

if(withdrawFlag) {

depositFlag=depositMoney(toAccountNo,amount);

}

if(depositFlag)

transferFlag=true;

else

depositMoney(fromAccountNo,amount);

return transferFlag;

}

public BankAccount[] getAccounts() {

return accounts;

}

public void setAccounts(BankAccount[] accounts) {

this.accounts = accounts;

}

public String getIFSC() {

return IFSC;

}

public void setIFSC(String iFSC) {

IFSC = iFSC;

}

public String getBankName() {

return bankName;

}

public void setBankName(String bankName) {

this.bankName = bankName;

}

@Override

public String toString() {

return "Bank [accounts=" + Arrays.toString(accounts) + ", IFSC=" + IFSC + ", bankName=" + bankName + "]";

}

}

**Interface:**

package overloading;

import overloading.BankAccount;

import overloading.Bank;

import overloading.InvalidAmountException;

import overloading.InvalidFundException;

public interface IBankServiceProvider {

BankAccount CheckAccount(String accountNo)throws InvalidAmountException;

double getBalance(String accountNo)throws InvalidAmountException;

boolean depositMoney(String accountNo,double amount)throws InvalidAmountException;

boolean withdrawMoney(String accountNo,double amount)throws InvalidAmountException,InvalidFundException;

boolean transferMoney(String fromAccount,String toAccountNo,double amount)throws InvalidAmountException,InvalidFundException;

}

**Exception1:**

package overloading;

import overloading.BankAccount;

import overloading.Bank;

public class InvalidAmountException extends Exception {

private String errorMsg="Invalid Account Number";

public InvalidAmountException(String errorMsg) {

super();

this.errorMsg = errorMsg;

}

public InvalidAmountException() {

super();

// TODO Auto-generated constructor stub

}

public String getMessage() {

return this.errorMsg;

}

}

**Exception2:**

package overloading;

import overloading.BankAccount;

import overloading.Bank;

public class InvalidFundException extends Exception {

private String errorMsg="Insufficient Balance";

public InvalidFundException(String errorMsg) {

super();

this.errorMsg = errorMsg;

}

public InvalidFundException() {

super();

// TODO Auto-generated constructor stub

}

public String getMessage() {

return this.errorMsg;

}

}

**BANKACCOUNTMAIN:**

**package** overloading;

**public** **class** BankAccountMain {

**public** **static** **void** main(String[] args) **throws** InvalidAmountException {

// **TODO** Auto-generated method stub

BankAccount[] myobj =**new** BankAccount[3];

myobj[0]=**new** BankAccount("100","Hari");

myobj[1]=**new** BankAccount("101","Mani");

myobj[2]=**new** BankAccount("102","Harish");

Bank hdfcBank=**new** Bank(myobj,"AshokNagar","HDFC1234");

BankAccount foundAccount=**null**;

**double** balance=0.0;

System.***out***.println("Initial Balance:"+hdfcBank);

**try** {

hdfcBank.withdrawMoney("100",1200);

hdfcBank.transferMoney("100","102",18);

}

**catch**(InvalidAmountException | InvalidFundException e) {

e.printStackTrace();

}

foundAccount=hdfcBank.CheckAccount("100");

balance=hdfcBank.getBalance("100");

hdfcBank.depositMoney("100",1500);

System.***out***.println("After Deposit & Withdraw from 1st to 3rd:"+foundAccount.toString());

**try** {

// System.out.println("Output by Try and catch");

System.***out***.println("Current Balance of 1st Account:"+hdfcBank.getBalance("100"));

System.***out***.println("Current Balance of 2nd Account:"+hdfcBank.getBalance("101"));

System.***out***.println("Current Balance of 3rd Account:"+hdfcBank.getBalance("102"));

}

**catch**(InvalidAmountException e) {

e.printStackTrace();

}

}

}

**QUESTION 7:**

**Address Class-**

**package** com.java.hashmap;

**public** **class** Address {

**public** **int** doorNo;

**public** String streetName;

**public** String cityName;

**public** **int** pinCode;

**public** **int** getDoorNo() {

**return** doorNo;

}

**public** **void** setDoorNo(**int** doorNo) {

**this**.doorNo = doorNo;

}

**public** String getStreetName() {

**return** streetName;

}

**public** **void** setStreetName(String streetName) {

**this**.streetName = streetName;

}

**public** String getCityName() {

**return** cityName;

}

**public** **void** setCityName(String cityName) {

**this**.cityName = cityName;

}

**public** **int** getPinCode() {

**return** pinCode;

}

**public** **void** setPinCode(**int** pinCode) {

**this**.pinCode = pinCode;

}

**public** Address(**int** doorNo, String streetName, String cityName, **int** pinCode) {

**super**();

**this**.doorNo = doorNo;

**this**.streetName = streetName;

**this**.cityName = cityName;

**this**.pinCode = pinCode;

}

**public** Address() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Address [doorNo=" + doorNo + ", streetName=" + streetName + ", cityName=" + cityName + ", pinCode="

+ pinCode + "]";

}

}

**Company Class:**

**package** com.java.hashmap;

**import** java.util.Arrays;

**import** com.java.hashmap.Employee;

**import** com.java.hashmap.Address;

**import** com.java.hashmap.ICompanyServiceProvider;

**import** java.util.HashMap;

**import** java.util.HashSet;

**public** **class** Company **implements** ICompanyServiceProvider{

**private** **int** companyID;

**public** Address address;

**public** Employee[] employee;

**public** **int** getCompanyID() {

**return** companyID;

}

**public** **void** setCompanyID(**int** companyID) {

**this**.companyID = companyID;

}

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

**public** Employee[] getEmployees() {

**return** employee;

}

**public** **void** setEmployees(Employee[] employees) {

**this**.employee = employees;

}

**public** HashMap<Integer, Employee> getEmployeesMap() {

**return** employeesMap;

}

**public** **void** setEmployeesMap(HashMap<Integer, Employee> employeesMap) {

**this**.employeesMap = employeesMap;

}

**public** Company(**int** companyID, Address address, Employee[] employees, HashMap<Integer, Employee> employeesMap) {

**super**();

**this**.companyID = companyID;

**this**.address = address;

**this**.employee = employees;

**this**.employeesMap = employeesMap;

}

**public** Company() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Company [companyID=" + companyID + ", address=" + address + ", employees=" + Arrays.*toString*(employee)

+ ", employeesMap=" + employeesMap + "]";

}

}

**Employee Class:**

package com.java.hashmap;

import java.util.HashMap;

public class Employee {

public int empID;

public String empName;

public String startDate;

private HashMap<Integer,Employee> employeesMap;

public Employee createEmp(Employee newEmp) {

Employee foundEmployee=null;

this.employeesMap.put(newEmp.getEmpID(),newEmp);

return foundEmployee;

}

public Employee readEmp(int empCode) {

Employee foundEmployee=null;

for(Employee emp:employee) {

if(emp.getEmpID()==empCode) {

foundEmployee=emp;

foundEmployee.getEmpID();

break;

}

}

return foundEmployee;

}

public Integer getEmpID() {

return empID;

}

public void setEmpID(int empID) {

this.empID = empID;

}

public String getStartDate() {

return startDate;

}

public void setStartDate(String startDate) {

this.startDate = startDate;

}

public String getEmpName() {

return empName;

}

public void setEmpName(String empName) {

this.empName = empName;

}

public Employee() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Employee [startDate=" + startDate + ", empName=" + empName + "]";

}

public Employee(int empID, String empName, String startDate) {

super();

this.empID = empID;

this.empName = empName;

this.startDate = startDate;

}

}

**Interface;**

**package** com.java.hashmap;

**import** com.java.hashmap.Employee;

**import** com.java.hashmap.Company;

**public** **interface** ICompanyServiceProvider {

**public** Employee createEmp(Employee newEmp);

**public** Employee readEmp(**int** empID);

**public** Employee updateEmp(**int** empID);

**public** Employee deleteEmp(**int** empID);

}

MainClass:

package com.java.hashmap;

import java.util.Arrays;

import java.util.HashMap;

import java.util.Map;

import java.util.Map.Entry;

import java.util.Set;

public class MainClass extends Company{

public static void main(String[] args) {

//Address address=new Address(12,"3rd Street","chennai",60095);

Company obj=new Company(12,address,employees,employeesMap);

Map<Integer,Employee> employees=new HashMap<Integer,Employee>();

Employee e1=new Employee("123","Hari","27 May");

Employee e2=new Employee("133","Hri","2 May");

employees.put(101,e1);

employees.put(102,e2);

//employees.put(103,"HAr");

//System.out.println(employees.get(101));

//System.out.println(employees.containsKey(103));

//Set<Entry<Integer,String>> mapentries=employees.entrySet();

//System.out.println(mapentries.toString());

for(Map.Entry<Integer,Employee> entry:employees.entrySet()){

int key=entry.getKey();

Employee b=entry.getValue();

//System.out.println((obj.getCompanyID()));

System.out.println(key+" Employee Details:");

System.out.println(entry.getValue());

}

**QUESTION 10 & 11:**

**Class Product:**

package com.java.tenth;

public class Product {

public int productID;

public String productName;

public double price;

public int quantityOnHand;

public int reorderLevel;

public int reorderQty;

public int getProductID() {

return productID;

}

public void setProductID(int productID) {

this.productID = productID;

}

public String getProductName() {

return productName;

}

public void setProductName(String productName) {

this.productName = productName;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

public int getQuantityOnHand() {

return quantityOnHand;

}

public void setQuantityOnHand(int quantityOnHand) {

this.quantityOnHand = quantityOnHand;

}

public int getReorderLevel() {

return reorderLevel;

}

public void setReorderLevel(int reorderLevel) {

this.reorderLevel = reorderLevel;

}

public int getReorderQty() {

return reorderQty;

}

public void setReorderQty(int reorderQty) {

this.reorderQty = reorderQty;

}

public Product(int productID, String productName, double price, int quantityOnHand, int reorderLevel, int reorderQty) {

super();

this.productID = productID;

this.productName = productName;

this.price = price;

this.quantityOnHand = quantityOnHand;

this.reorderLevel = reorderLevel;

this.reorderQty = reorderQty;

}

public Product() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Product [productID=" + productID + ", productName=" + productName + ", price=" + price + ", quantityOnHand="

+ quantityOnHand + ", reorderLevel=" + reorderLevel + ", reorderQty=" + reorderQty + "]";

}

}

**Class Stores:**

package com.java.tenth;

import java.util.Arrays;

public class Stores implements StoreServiceProvider {

public Product[] products;

public Product[] getProducts() {

return products;

}

public void setProducts(Product[] products) {

this.products = products;

}

public Stores() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Stores [products=" + Arrays.toString(products) + "]";

}

public Stores(Product[] products) {

super();

this.products=products;

}

public double sellItem(int productCode,int qtyRequired)throws ProductNotFoundException {

double billAmount=0.0;

Product foundProduct=null;

for(Product item:this.products) {

if(item.getProductID()==productCode) {

foundProduct=item;

break;

}

}

if(foundProduct!=null) {

if(foundProduct.getQuantityOnHand()>=qtyRequired){

foundProduct.setQuantityOnHand(foundProduct.getQuantityOnHand()-qtyRequired);

billAmount=foundProduct.getPrice()\*qtyRequired;

}

}

else

throw new ProductNotFoundException();

return billAmount;

}

@Override

public boolean updateStock(int productCode, int arrivedQty)throws ProductNotFoundException {

Product foundProduct=null;

for(Product item:this.products) {

if(item.getProductID()==productCode) {

foundProduct=item;

break;

}

}

if(foundProduct!=null) {

foundProduct.setQuantityOnHand(foundProduct.getQuantityOnHand()+arrivedQty);

}

else

throw new ProductNotFoundException();

return true;

}

}

**ExceptionClass:**

package com.java.tenth;

public class ProductNotFoundException extends Exception {

private String errorMsg="Product Not Found";

public ProductNotFoundException() {

super();

}

public ProductNotFoundException(String errorMsg) {

super();

this.errorMsg = errorMsg;

}

public String getMessage() {

return this.errorMsg;

}

}

**Interface:**

package com.java.tenth;

public interface StoreServiceProvider {

public double sellItem(int productCode,int qtyRequired) throws ProductNotFoundException;

public boolean updateStock(int productCode,int arrivedQty) throws ProductNotFoundException;

}

**Main Class:**

package com.java.tenth;

import java.util.Arrays;

public class MainClass {

public static void main(String[] args) throws ProductNotFoundException {

// TODO Auto-generated method stub

Product[] products=new Product[3];

products[0]=new Product(123,"Dhall",120.00,100,10,50);

products[1]=new Product(124,"Pencil",20.00,20,10,50);

products[2]=new Product(125,"Soap",50.00,10,10,50);

Stores str=new Stores();

double billValue=str.sellItem(123,5);

boolean arrived=str.updateStock(125,10);

System.out.println("BillValue:" +billValue);

System.out.println("ArrivedQty:"+arrived);

System.out.println(Arrays.toString(str.getProducts()));

}

}