**JAVA 8 Features & I/O & Reflection**

**Q-1. Write a program to copy and paste a file to other location**

**import** org.apache.commons.io.FileUtils;

**import** java.io.File;

**public class** Q1 {

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

File source = **new** File(**"/home/prachi/Desktop/Document"**);

File dest = **new** File(**"/home/prachi/Documents"**);

**try** {

FileUtils.*copyFileToDirectory*(source, dest);

System.***out***.println(**"File copied successfully"**);

}

**catch** (Exception e) {

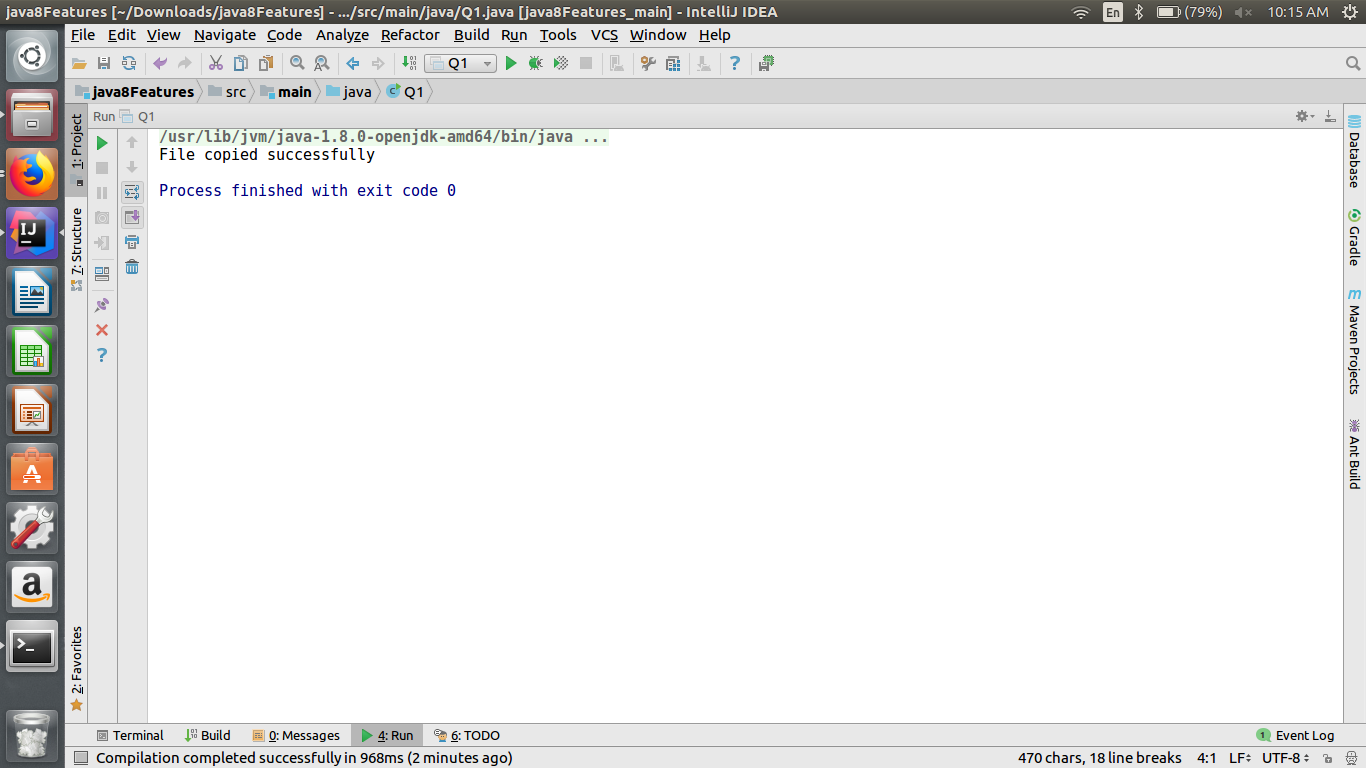
e.printStackTrace();

}

}

}

**OUTPUT:-**



**Q-2. Write a program to serialize a class Employee with property name and age. Then deserialize it and print values.**

**import java.io.\*;**

**class Employee implements Serializable{**

**private String name;**

**private int id;**

**private int salary;**

**private String department;**

**public int getSalary() {**

**return salary;**

**}**

**public void setSalary(int salary) {**

**this.salary = salary;**

**}**

**public String getDepartment() {**

**return department;**

**}**

**public void setDepartment(String department) {**

**this.department = department;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public int getId() {**

**return id;**

**}**

**public void setId(int id) {**

**this.id = id;**

**}**

**}**

**public class Q2 {**

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

**Employee e = new Employee();**

**e.setName("Prachi Julka");**

**e.setId(101);**

**e.setDepartment("Accounts");**

**e.setSalary(100000);**

**FileOutputStream fileOut =**

**new FileOutputStream("/home/prachi/Documents/employee.ser");**

**ObjectOutputStream out = new ObjectOutputStream(fileOut);**

**out.writeObject(e);**

**out.close();**

**fileOut.close();**

**System.*out*.println("Serialized data is saved in /home/prachi/Documents/employee.ser");**

**FileInputStream fis=new FileInputStream("/home/prachi/Documents/employee.ser");**

**ObjectInputStream ois = new ObjectInputStream(fis);**

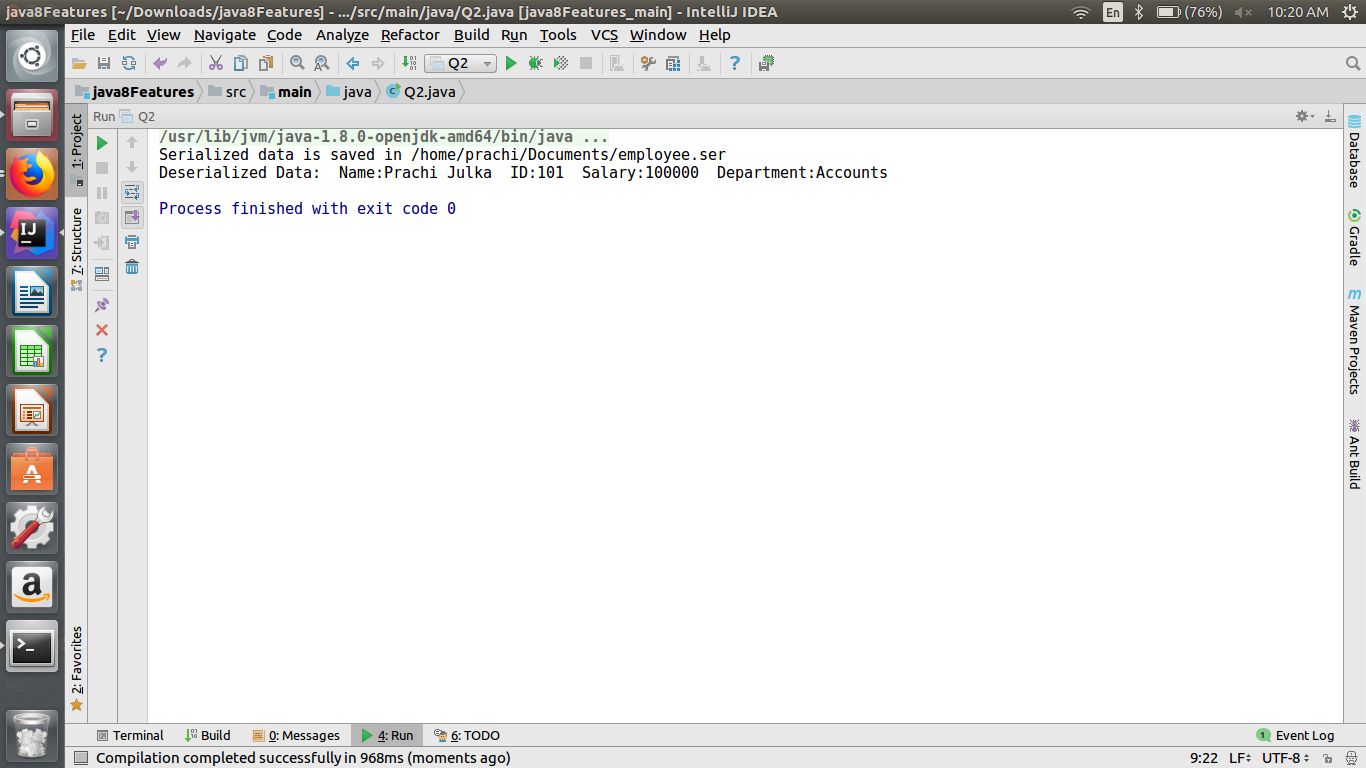
**Employee e1 =(Employee)ois.readObject();**

**System.*out*.println("Deserialized Data: Name:"+e1.getName()+" ID:"+e1.getId() +" Salary:"+ e1.getSalary()+" Department:"+ e1.getDepartment());**

**}**

**}**

**OUTPUT:-**

****

**Q-3 Write a program to read all field and method names of a class using reflection.**

**import java.lang.reflect.Field;**

**import java.lang.reflect.Method;**

**import java.util.Scanner;**

**public class Q3 {**

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

**Student e=new Student();**

**Class clazz = e.getClass();**

**Method methodset= clazz.getDeclaredMethod("setDetails", new Class[]**

**{});**

**methodset.setAccessible(true);**

**methodset.invoke(e);**

**Field name=e.getClass().getDeclaredField("name");**

**name.setAccessible(true);**

**Object valueName=name.get(e);**

**Field rollno=e.getClass().getDeclaredField("rollno");**

**rollno.setAccessible(true);**

**Object value1=rollno.get(e);**

**System.*out*.println("Name: "+valueName);**

**System.*out*.println("Rollno: "+value1);**

**}**

**}**

**class Student{**

**private String name;**

**private int rollno;**

**Scanner sc=new Scanner(System.*in*);**

**private void setDetails(){**

**System.*out*.println("Enter your name");**

**name=sc.next();**

**System.*out*.println("Enter your rollno");**

**rollno=sc.nextInt();**

**}**

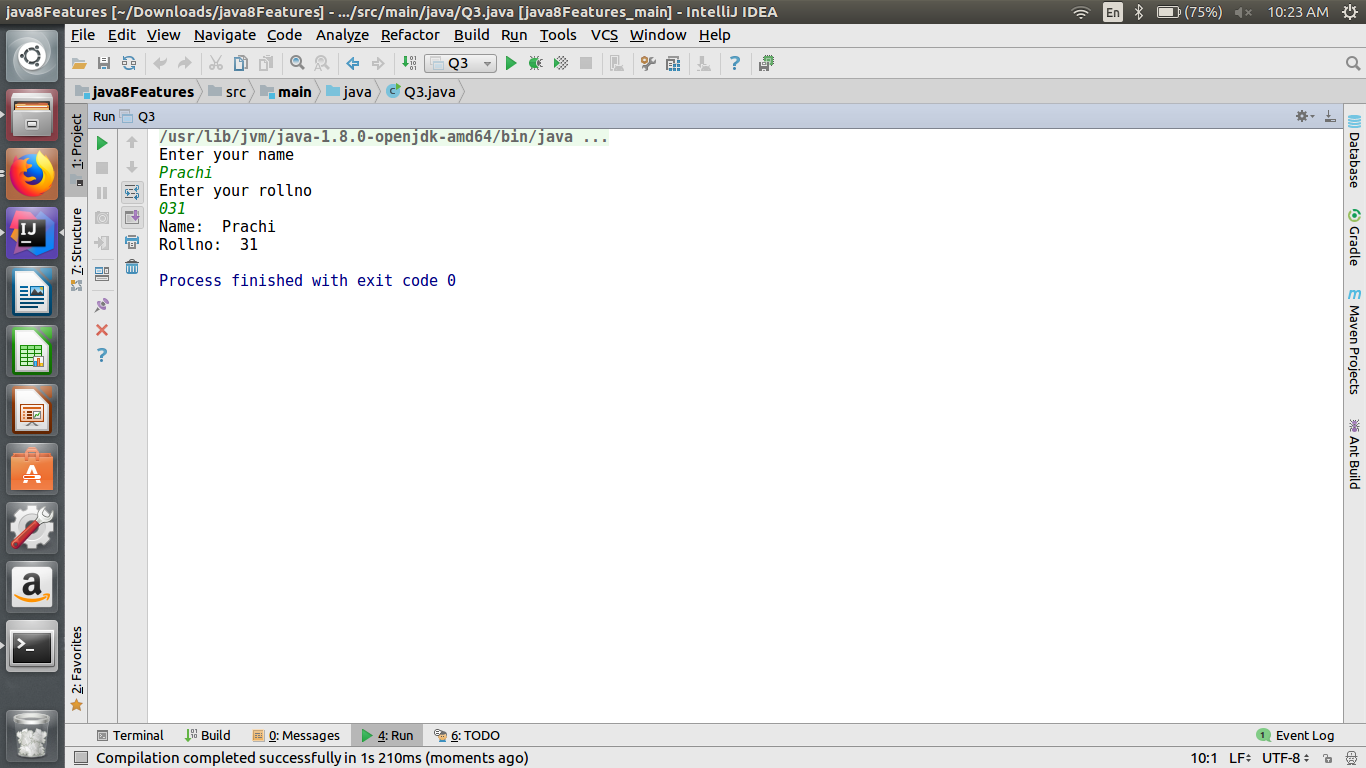
**private void getDetails(){**

**System.*out*.println("NAME: "+name);**

**System.*out*.println("Roll Number: "+rollno);**

**}**

**}**

**OUTPUT:-**

**Q-4 Write a program to print a table of any number using lambda expression.**

**public class Q4 {**

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

**Q4 obj=new Q4();**

**Demo t =(int a) -> {for(int i=1;i<=10;i++) System.*out*.println(a +"X"+ i+ "=" + a \* i);};**

**t.table(5);**

**}**

**@FunctionalInterface**

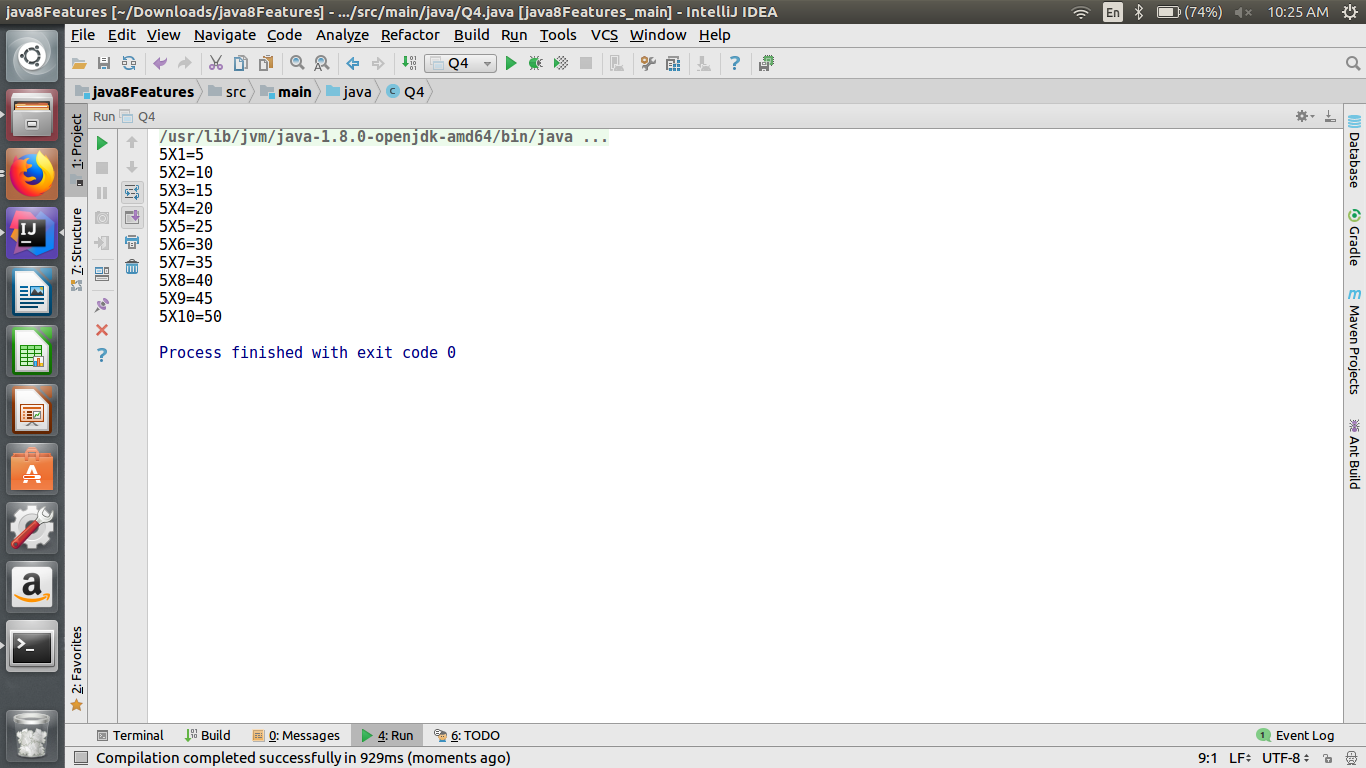
**interface Demo{**

**void table(int a);**

**}**

**}**

**OUTPUT:-**

****

**Q-5 Write a program having interface with default and static methods and call them also in your main method.**

**public class Q5 implements New {**

**void demo(){**

**System.*out*.println("Hello from Q5 mehtod");**

**}**

**@Override**

**public void demoDefault(){**

**System.*out*.println("Hello from overrided default method");**

**}**

**public void demoStatic(){**

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

**}**

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

**Q5 obj=new Q5();**

**obj.demo();**

**obj.demoDefault();**

**New.*demoStatic*();**

**}**

**}**

**interface New{**

**default void demoDefault(){**

**System.*out*.println("Hello from default method");**

**}**

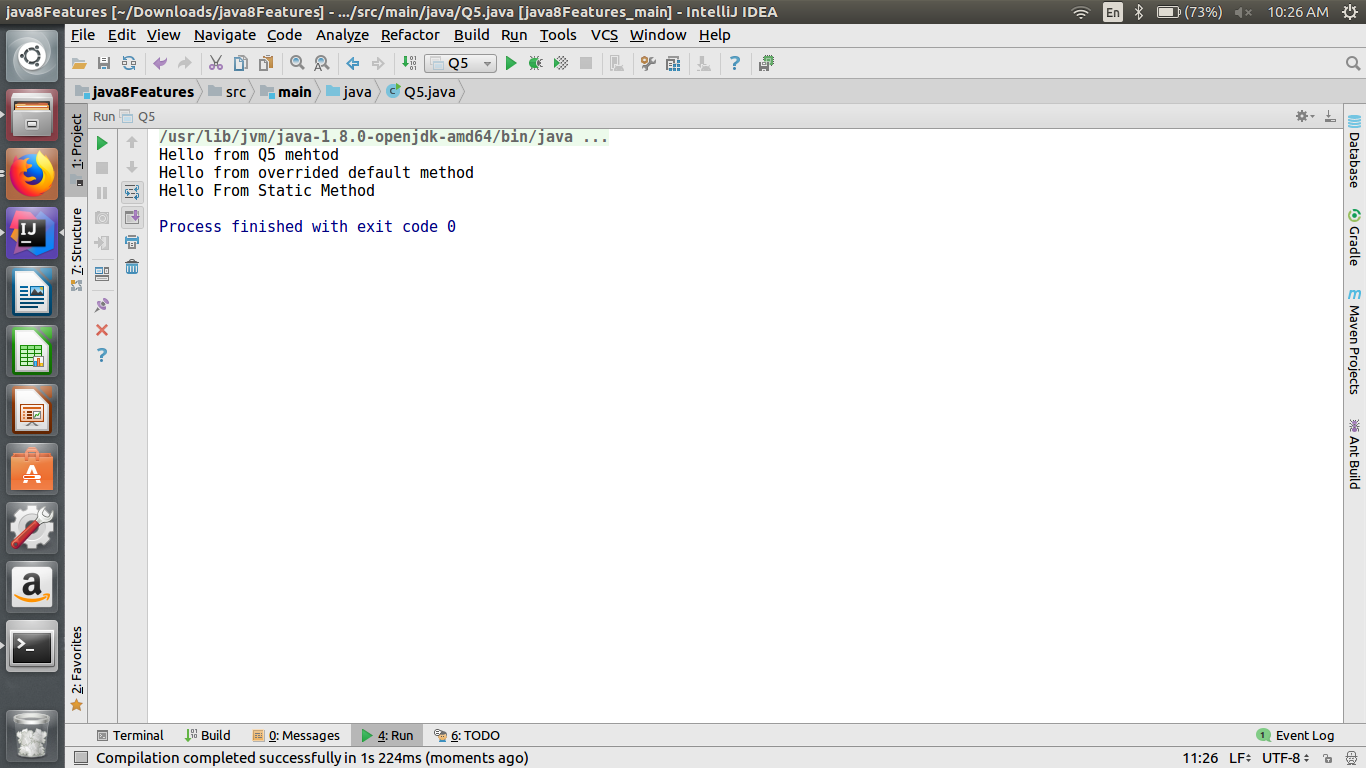
**static void demoStatic(){**

**System.*out*.println("Hello From Static Method");**

**}**

**}**

**OUTPUT:-**

****

**Q-6 Create a class Employee with name and age. Then write a program to print all those employees whose name start with n and age is greater than 24.**

**public class Q6 {**

**String name;**

**int age;**

**public Q6(String name,int age){**

**this.name=name;**

**this.age=age;**

**}**

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

**Q6[]obj=new Q6[5];**

**obj[0]=new Q6("Prachi",20);**

**obj[1]=new Q6("Nancy",25);**

**obj[2]=new Q6("Ankita",26);**

**obj[3]=new Q6("Ananya",24);**

**obj[4]=new Q6("Anjali",19);**

**System.*out*.println("Details of employees whose name start with n and age is greater than 24");**

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

**for(int i=0;i<5;i++){**

**if(obj[i].age>24 && obj[i].name.charAt(0)=='N'){**

**System.*out*.println(obj[i].name+" "+obj[i].age);**

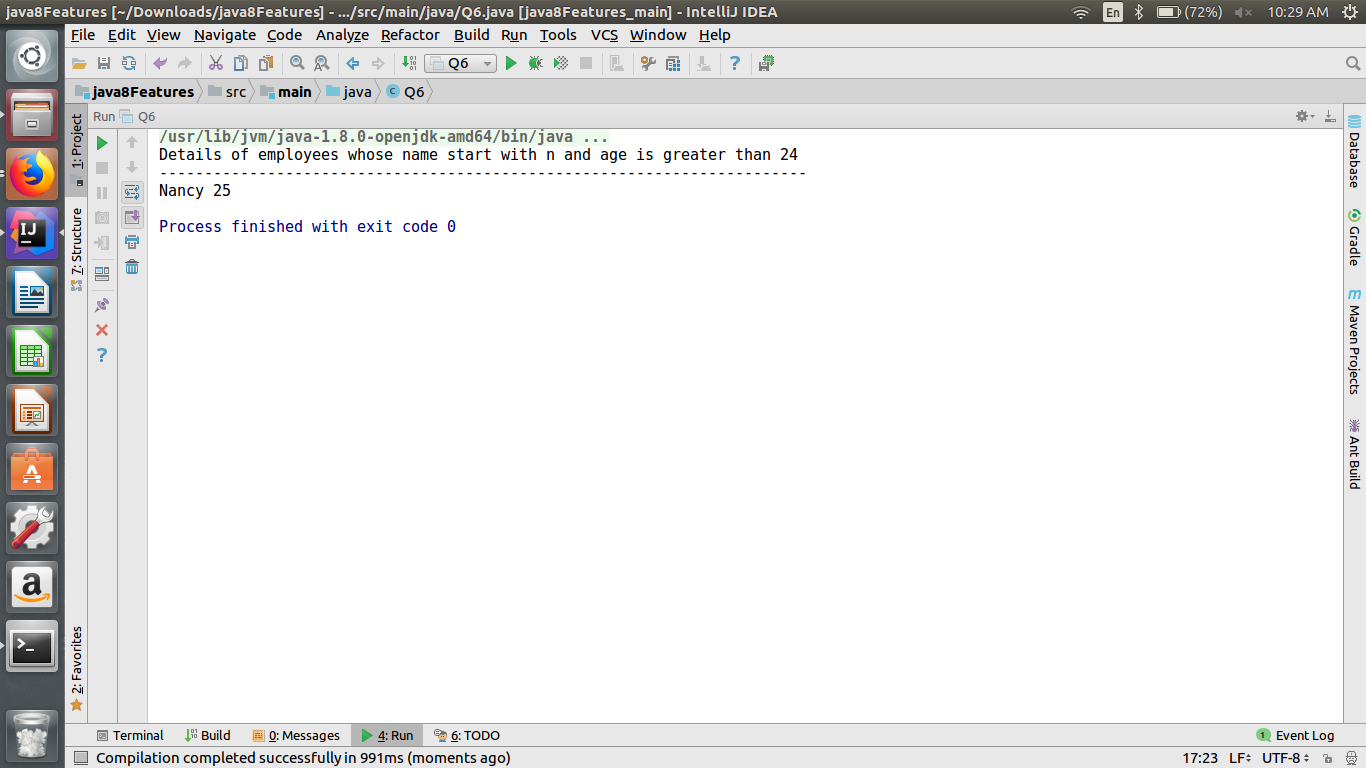
**}**

**}**

**}**

**}**

**OUTPUT:-**

****