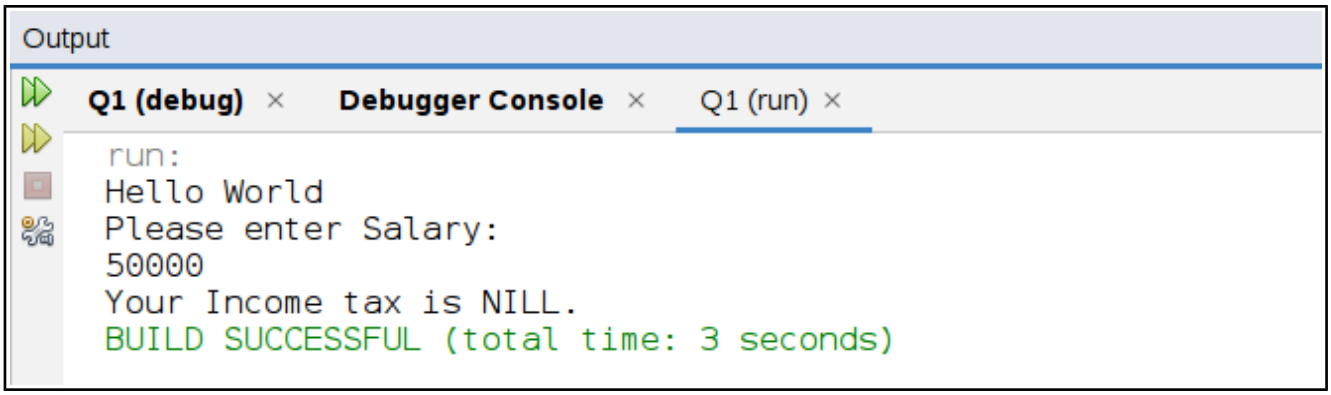
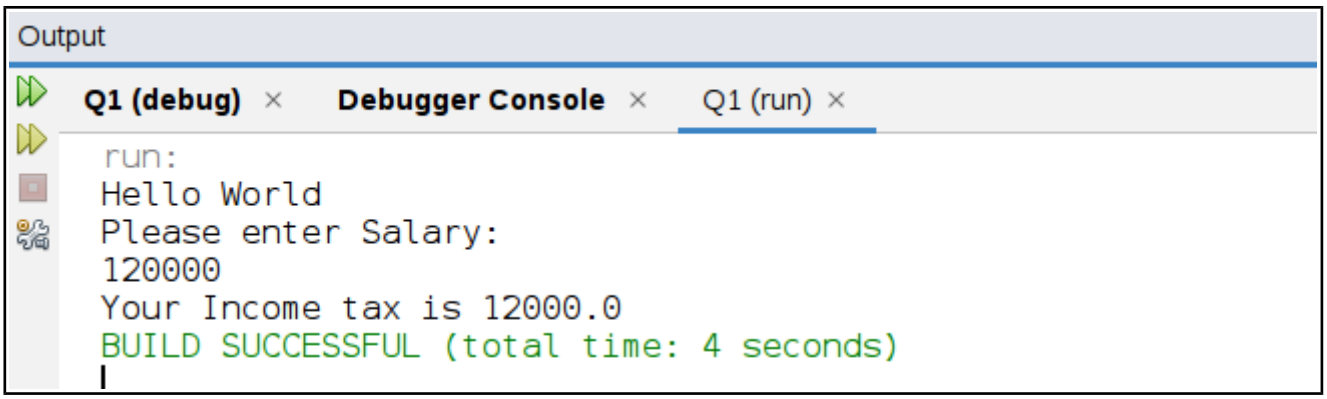
**Question 1: Write a Java program which initialization of earning of an employee. The program should calculate the income tax to be paid by the employee as per the criteria given below.**

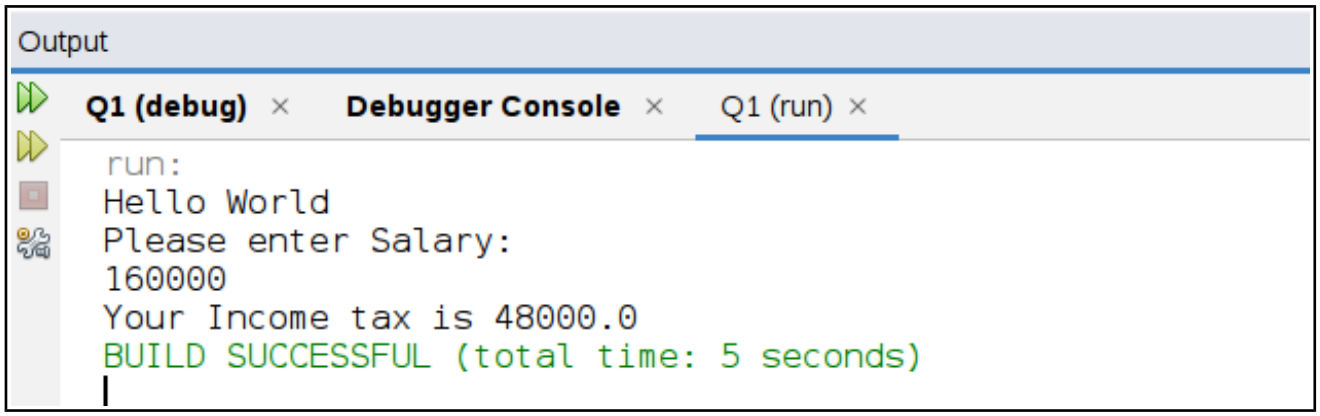
**Slab Rate IT Rate**
**Upto Rs. 50,000 Nil**
**Upto Rs.60,000 10% on additional amount**
**Upto Rs.1,50,000 20% on additional amount**
**Above Rs. 1,50,000 30% on additional amount**
**Result :- Income tax is .....**

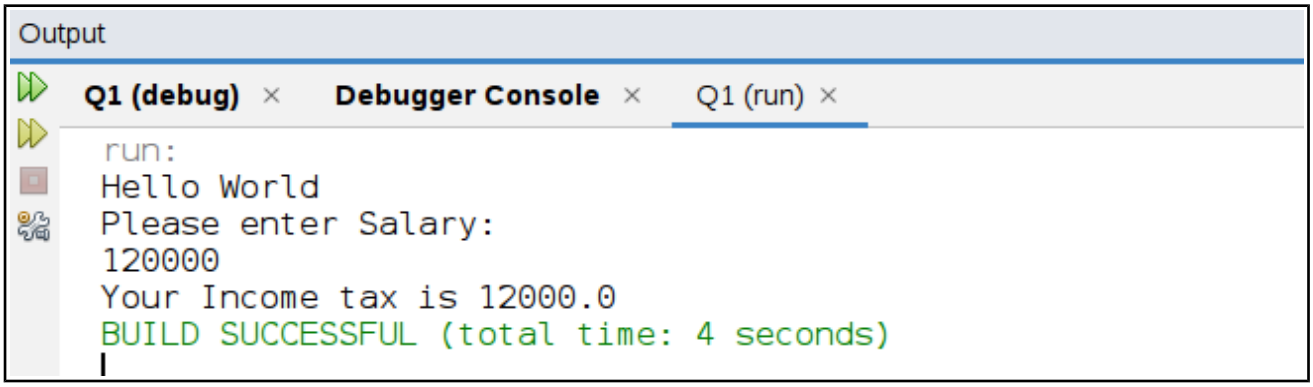
---

```
import java.security.DrbgParameters;
import java.util.*;

/**
 * @author root
 */
public class Sallarytax {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World");
        System.out.println("Please enter Salary:");
        Scanner sc=new Scanner(System.in);
        Double salary =sc.nextDouble();
        if(salary<=50000)
        {
            System.out.println("Your Income tax is NILL.");
        }
        else if(salary>=60000 && salary<150000)
        {
            salary=(10*salary)/100;
            System.out.println("Your Income tax is "+salary);
        }
        else if(salary>=150000 && salary<60000)
        {
            salary=(20*salary)/100;
            System.out.println("Your Income tax is "+salary);
        }
        else if(salary>150000)
        {
            salary=(30*salary)/100;
            System.out.println("Your Income tax is "+salary);
        }
    }

}
```


Output
Q1 (debug)
Debugger Console
Q1 (run)
run:
Hello World
Please enter Salary:
50000
Your Income tax is NILL.
BUILD SUCCESSFUL (total time: 3 seconds)


Output
Q1 (debug)
Debugger Console
Q1 (run)
run:
Hello World
Please enter Salary:
120000
Your Income tax is 12000.0
BUILD SUCCESSFUL (total time: 4 seconds)


Output
Q1 (debug)
Debugger Console
Q1 (run)
run:
Hello World
Please enter Salary:
160000
Your Income tax is 48000.0
BUILD SUCCESSFUL (total time: 5 seconds)


Output
Q1 (debug)
Debugger Console
Q1 (run)
run:
Hello World
Please enter Salary:
120000
Your Income tax is 12000.0
BUILD SUCCESSFUL (total time: 4 seconds)

**Question 2: Write method of date class to support the following:**
**a) Method for validating that the integer -representing month is between 1 & 12 and checking that the day part of the date objects is within the correct range of month.**
**b) Obtaining the next day from a given date.**

---

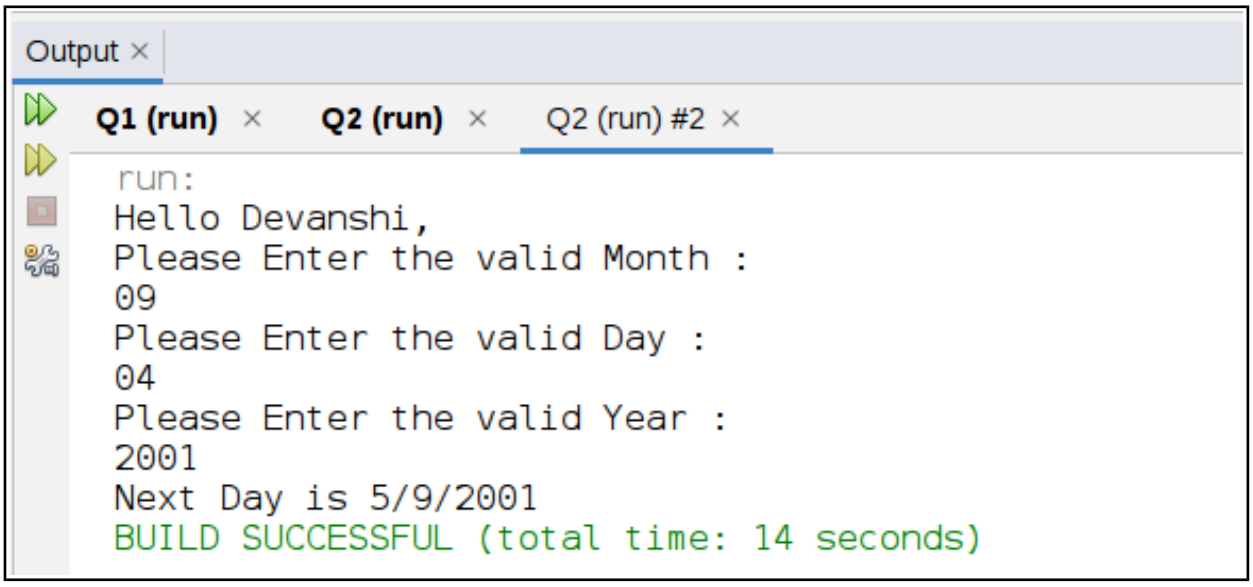
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package q2;

import java.util.*;


public class NxtDate{
    public static void main(String[] args) {
        System.out.println("Hello Dear");
        Scanner sc=new Scanner(System.in);
        System.out.println("Please Enter the valid Month :");
        int month=sc.nextInt();
        System.out.println("Please Enter the valid Day :");
        int day=sc.nextInt();
        System.out.println("Please Enter the valid Year :");
        int year=sc.nextInt();
        if(month==1 && day>0 && day<=31 && year>999 && year<=9999)
        {
            if(day==31)
            {
                day=1;
                month=month+1;
                System.out.println("Next Day is "+day+"/"+month+"/"+year);
            }
            else
            {
                day=day+1;
                System.out.println("Next Day is "+day+"/"+month+"/"+year);
            }
        }
        else if(month==2 && day>0 && day<=28 && year>999 && year<=9999)
        {
            if(day==28)
            {
                day=1;
                month=month+1;
                System.out.println("Next Day is "+day+"/"+month+"/"+year);
            }
```

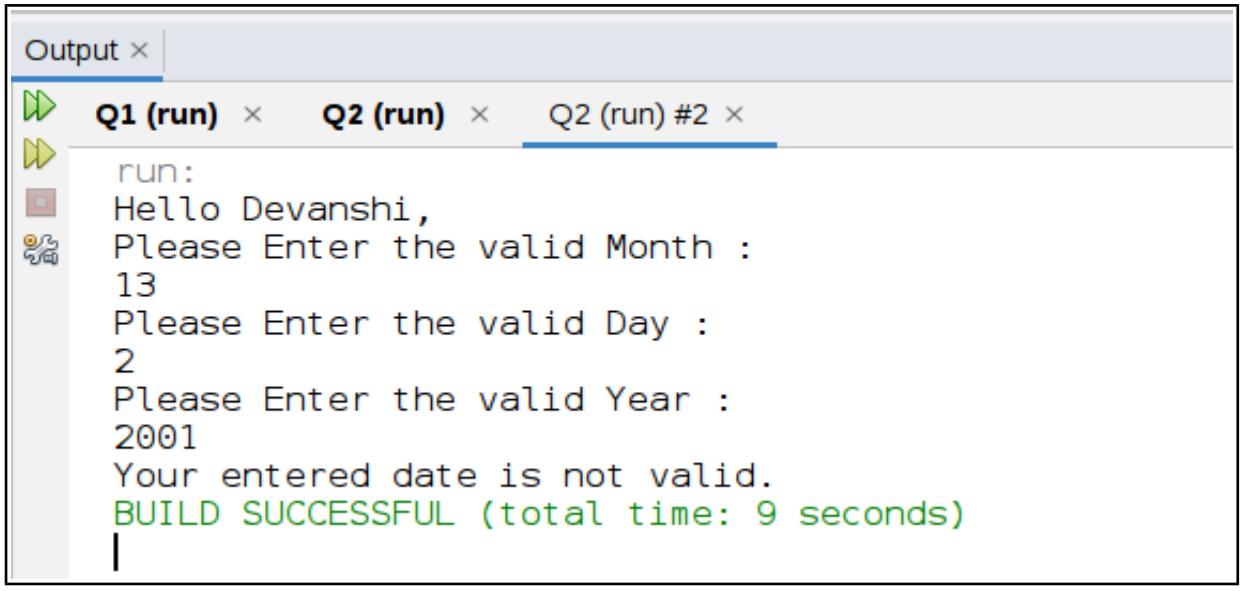
```
    else
    {
       day=day+1;
       System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
else if(month==3 && day>0 && day<=31 && year>999 && year<=9999)
{
    if(day==31)
    {
       day=1;
       month=month+1;
       System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
    else
    {
       day=day+1;
       System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
else if(month==4 && day>0 && day<=30 && year>999 && year<=9999)
{
    if(day==30)
    {
       day=1;
       month=month+1;
       System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
    else
    {
       day=day+1;
       System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
else if(month==5 && day>0 && day<=31 && year>999 && year<=9999)
{
    if(day==31)
    {
       day=1;
       month=month+1;
       System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
    else
    {
       day=day+1;
       System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
else if(month==6 && day>0 && day<=30 && year>999 && year<=9999)
```

```
{
   if(day==30)
   {
      day=1;
      month=month+1;
      System.out.println("Next Day is "+day+"/"+month+"/"+year);
   }
   else
   {
      day=day+1;
      System.out.println("Next Day is "+day+"/"+month+"/"+year);
   }
}
else if(month==7 && day>0 && day<=31 && year>999 && year<=9999)
{
   if(day==31)
   {
      day=1;
      month=month+1;
      System.out.println("Next Day is "+day+"/"+month+"/"+year);
   }
   else
   {
      day=day+1;
      System.out.println("Next Day is "+day+"/"+month+"/"+year);
   }
}
else if(month==8 && day>0 && day<=31 && year>999 && year<=9999)
{
   if(day==31)
   {
      day=1;
      month=month+1;
      System.out.println("Next Day is "+day+"/"+month+"/"+year);
   }
   else
   {
      day=day+1;
      System.out.println("Next Day is "+day+"/"+month+"/"+year);
   }
}
else if(month==9 && day>0 && day<=30 && year>999 && year<=9999)
{
   if(day==30)
   {
      day=1;
      month=month+1;
      System.out.println("Next Day is "+day+"/"+month+"/"+year);
   }
```

```
    else
    {
        day=day+1;
        System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
else if(month==10 && day>0 && day<=31 && year>999 && year<=9999)
{
    if(day==31)
    {
        day=1;
        month=month+1;
        System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
    else
    {
        day=day+1;
        System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
else if(month==11 && day>0 && day<=30 && year>999 && year<=9999)
{
    if(day==30)
    {
        day=1;
        month=month+1;
        System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
    else
    {
        day=day+1;
        System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
else if(month==12 && day>0 && day<=31 && year>999 && year<=9999)
{
    if(day==31)
    {
        day=1;
        month=1;
        year=year+1;
        System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
    else
    {
        day=day+1;
        System.out.println("Next Day is "+day+"/"+month+"/"+year);
    }
}
```

```
        else
        {
            System.out.println("Your entered date is not valid.");
        }
    }

}
```



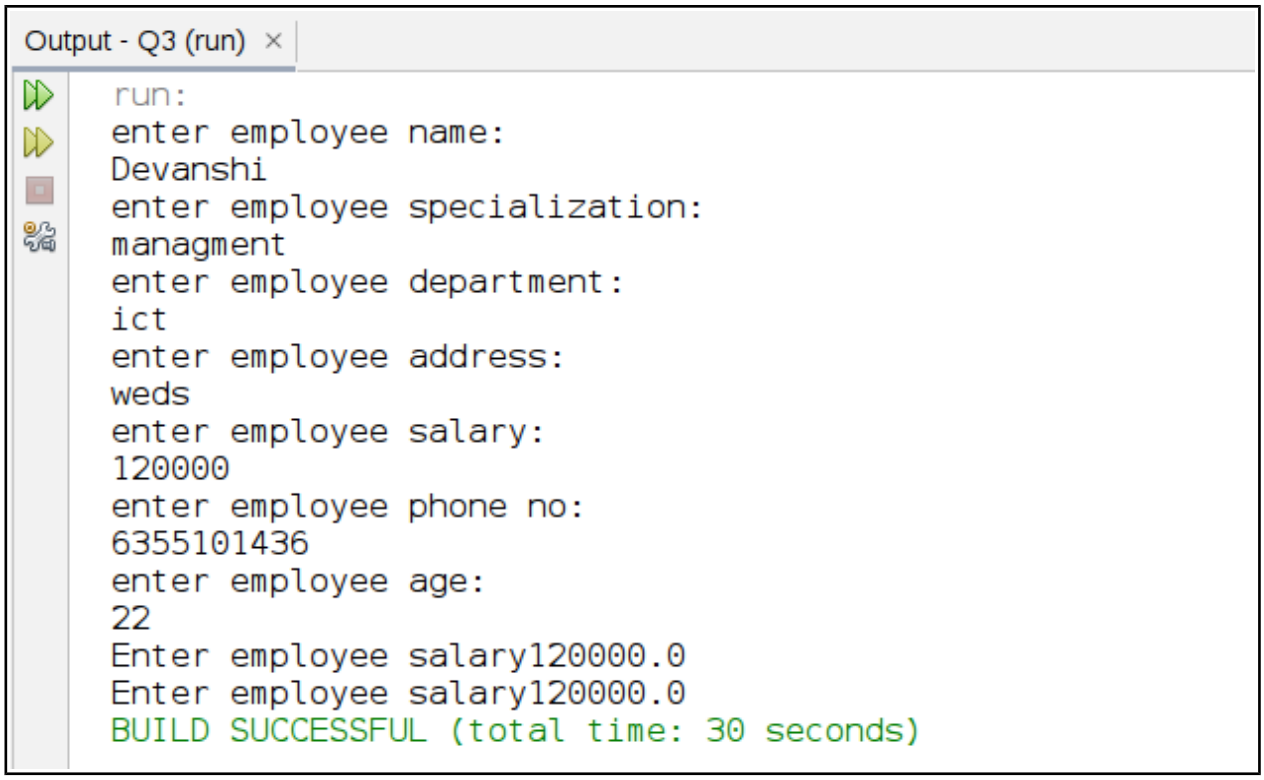



**Question 3: Create a class named 'Member' having the following members: Data members**
**1 - Name**
**2 - Age**
**3 - Phone number**
**4 - Address**
**5 - Salary**
**It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively.**
**Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.**

---

```
package q3;

import java.util.Scanner;

class Member{

    String name,address;
    float salary;
    String phone;
    int age;

    public Member(String name, String address, float salary,String phone, int age) {
        this.name = name;
        this.address = address;
        this.salary = salary;
        this.phone = phone;
        this.age = age;
    }

    void printsalary(){
        System.out.println("Enter Employee salary"+salary);
    }

}
class employee extends Member{

   String specialization;

    public employee(String specialization, String name, String address, float salary, String phone, int age)
{
        super(name, address, salary, phone, age);
        this.specialization = specialization;
    }
}
```

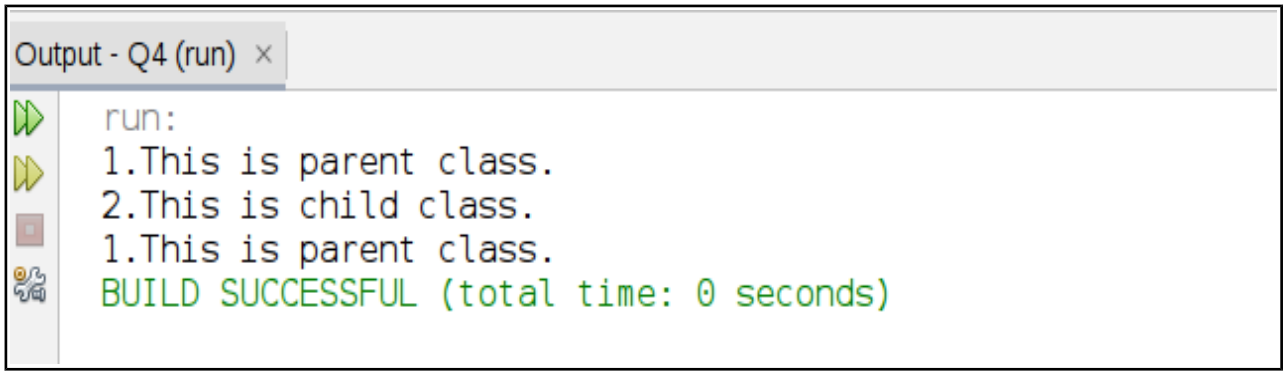
```
class manager extends Member{

    String department;

    public manager(String department, String name, String address, float salary, String phone, int age) {
        super(name, address, salary, phone, age);
        this.department = department;
    }



}
public class Q3{

    public static void main(String[] args) {
        // TODO code application logic here
        Scanner input = new Scanner(System.in);
        System.out.println("enter employee name: ");
        String name=input.nextLine();
        System.out.println("enter employee specialization: ");
        String sp=input.nextLine();
        System.out.println("enter employee department: ");
        String dp=input.nextLine();
        System.out.println("enter employee address: ");
        String address=input.nextLine();
        System.out.println("enter employee salary: ");
        float salary=input.nextFloat();
        input.nextLine();
        System.out.println("enter employee phone no: ");
        String ph=input.nextLine();
        System.out.println("enter employee age: ");
        int age=input.nextInt();
        employee e = new employee(sp,name,address,salary,ph,age);
        e.printsalary();
        manager m = new manager(dp,name,address,salary,ph,age);
        m.printsalary();

    }

}
```


Output - Q3 (run)
run:
enter employee name:
Devanshi
enter employee specialization:
managment
enter employee department:
ict
enter employee address:
weds
enter employee salary:
120000
enter employee phone no:
6355101436
enter employee age:
22
Enter employee salary120000.0
Enter employee salary120000.0
BUILD SUCCESSFUL (total time: 30 seconds)

**Question 4: Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call**
**1 - method of parent class by object of parent class**
**2 - method of child class by object of child class**
**3 - method of parent class by object of child class**

---

```
package q4;

/**
 *
 * @author root
 */
public class Q4 {

    public static void main(String[] args) {
        parent parentobj=new parent();
        child childobj=new child();
        parentobj.parentmethod();
        childobj.childmethod();
        childobj.parentmethod();
    }

}
class parent{
    public void parentmethod()
    {
        System.out.println("1.This is parent class.");
    }
}

class child extends parent{
    public void childmethod()
    {
        System.out.println("2.This is child class.");
    }
}
```

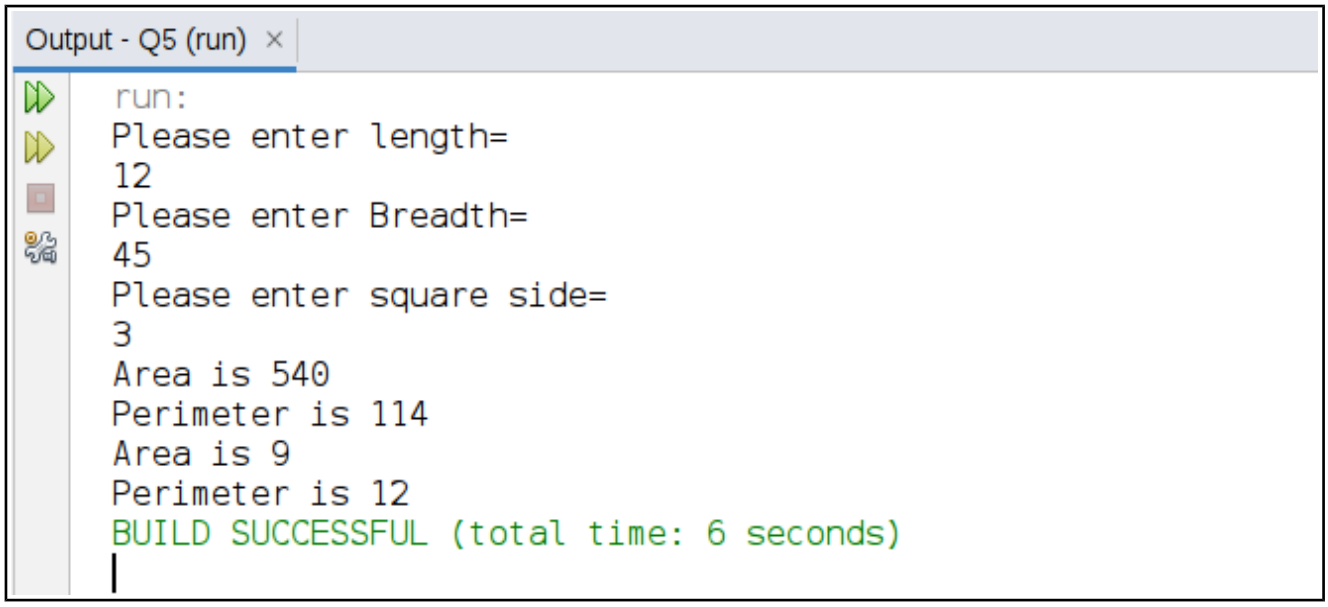


**Question 5: Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.**

---

```
package q5;

import java.util.*;

/**
 *
 * @author root
 */
public class Q5{

    public static void main(String[] args) {
        System.out.println("Please enter length=");
        Scanner sc=new Scanner(System.in);
        int l=sc.nextInt();
        System.out.println("Please enter Breadth=");
        int b=sc.nextInt();
        sc.nextLine();
        System.out.println("Please enter square side=");
        int ss=sc.nextInt();
        rectangle obj=new rectangle(l, b);
        square s=new square(ss);
        obj.print_area();
        obj.print_parimeter();
        s.print_area();
        s.print_parimeter();
    }

}
class rectangle{
    int length,breadth;
    public rectangle(int l,int b)
    {
        length=l;
        breadth=b;
    }
    void print_area()
    {
        System.out.println("Area is "+(length*breadth));
    }
    void print_parimeter()
    {
```

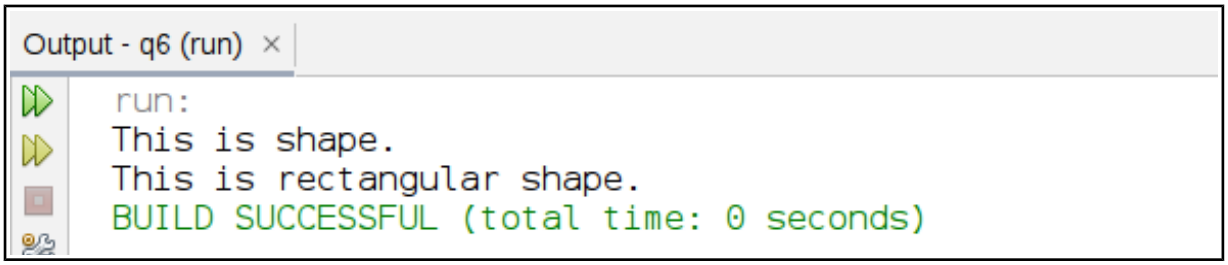
```
        System.out.println("Perimeter is "+(2*(length+breadth)));
    }
}
class square extends rectangle
{
    public square(int s)
    {
        super(s, s);
    }
}
```



**Question 6: Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.**

---

```
package q6;

public class Q6 {
    public static void main(String[] args) {
        square sobj=new square();
        sobj.shamethod();
        sobj.recmethod();
    }

}
class shape{
    public void shamethod()
    {
        System.out.println("This is shape.");
    }
}
class rectangle extends shape{
    public void recmethod()
    {
        System.out.println("This is rectangular shape.");
    }
}
class square extends rectangle{
    public void squmethod()
    {
        System.out.println("This is square is a rectangle.");
    }
}
class circle extends shape{
    public void cirmethod()
    {
        System.out.println("This is circular shape.");
    }
}
```

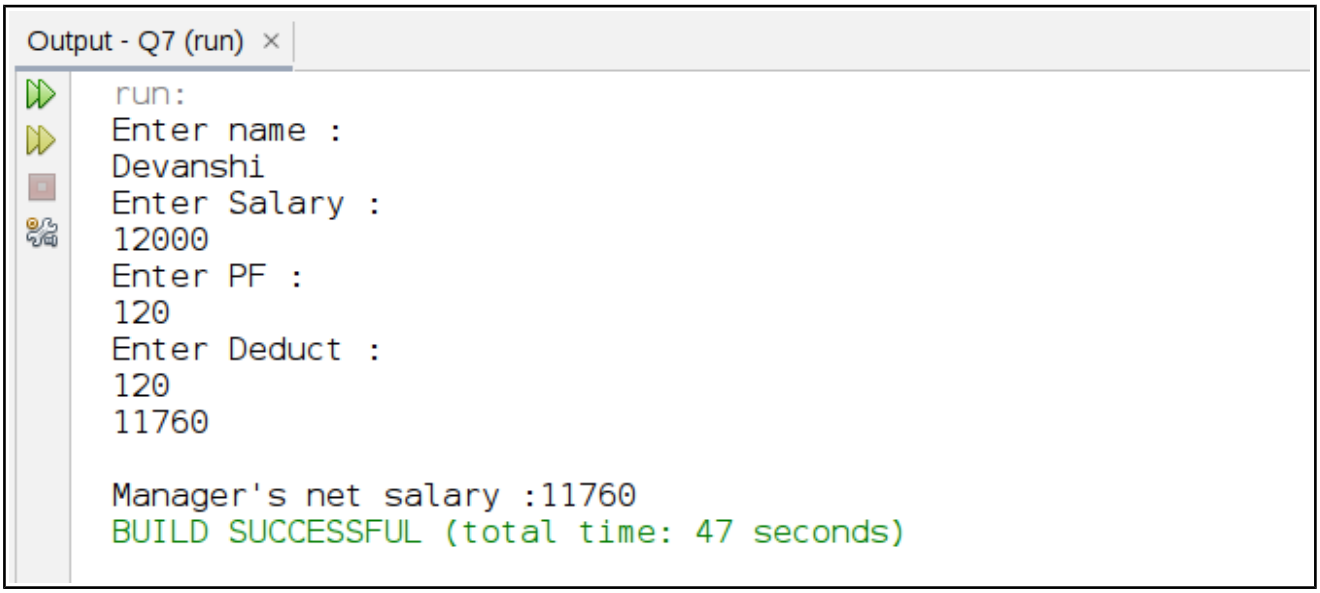


**Question 7: Create an abstract class employee, having it properties and abstract function for calculating net salary and displaying the information. Drive manager and clerk class from this abstract class and implement the abstract method net salary and override the display method.**

---

```
package q7;
import java.util.Scanner;
/**
* @author root
*/
abstract class employee{
    int netsal,sal,pf,dec;
    void netsal(){
        netsal=sal-pf-dec;
        System.out.println(netsal);
    }


    void getdata(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter name :");
        String name=sc.nextLine();
        System.out.println("Enter Salary :");
        sal=sc.nextInt();
        System.out.println("Enter PF :");
        pf=sc.nextInt();
        System.out.println("Enter Deduct :");
        dec=sc.nextInt();
        }
    abstract void display();
}

class manager extends employee{
    @Override
    void display() {
    System.out.println();
    System.out.println("Manager's net salary :" +netsal);
    }
}

class clerk extends employee{
    @Override
    void display() {
        System.out.println();
        System.out.println("Clerk's net salary :" +netsal);
    }
}
```

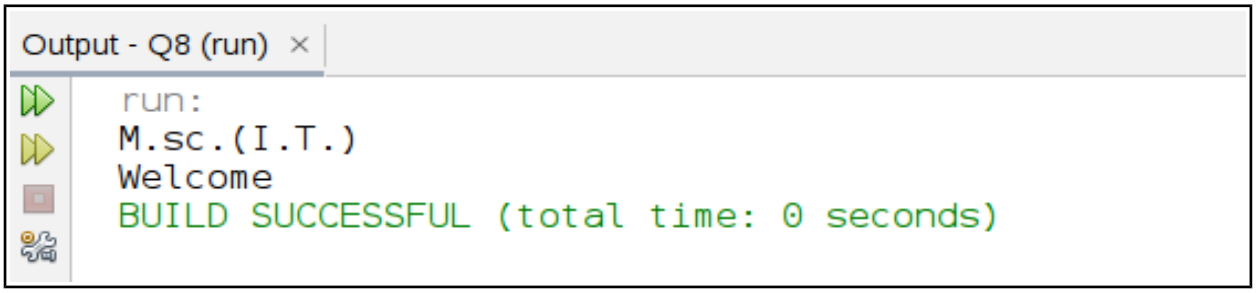
```
public class Q7 {
   public static void main(String args[]){
      manager m =new manager();
      m.getdata();
      m.netsal();
      m.display();
   }
}
```



**Question 8: Write a java program to create a two threads, one prints “M.s.c(I.T)” and other prints “Welcome”.**

---

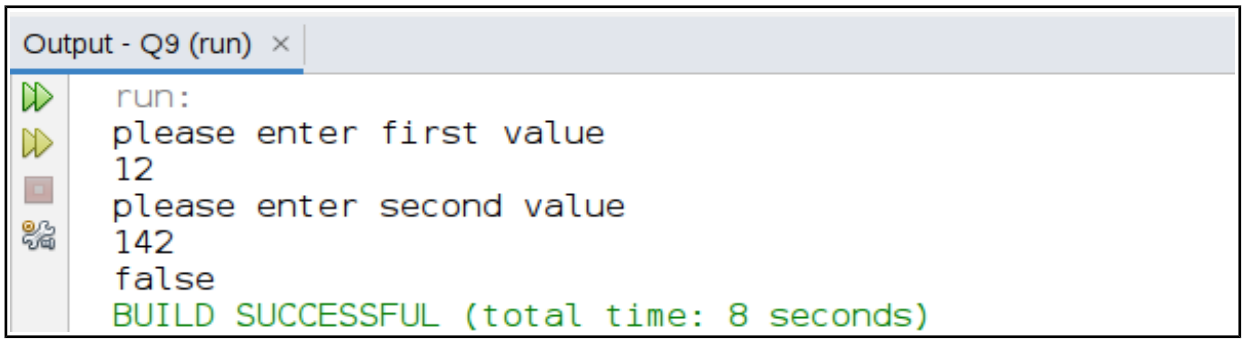
```
package q8;

/**
 * @author root
 */

public class Q8 {

    public static void main(String[] args) {
        Thread t1=new Thread(new Runnable()
        {
            public void run(){
                System.out.println("M.sc.(I.T.)");
            }
        });
        Thread t2=new Thread(new Runnable() {
            @Override
            public void run() {
                System.out.println("Welcome");
            }
        });
        t1.start();
        t2.start();
    }

}
```


Output - Q8 (run)
run:
M.sc.(I.T.)
Welcome
BUILD SUCCESSFUL (total time: 0 seconds)

**Question 9: Create a class Student with following operations**
**1)create parameterized constructor to initialize the objects**
**2) create a function isEqual() to check whether the two objects are equal or not which returns the Boolean value and gets two objects**
**3) print the result in main method if objects are equals or not (take variables as your assumption)**

---

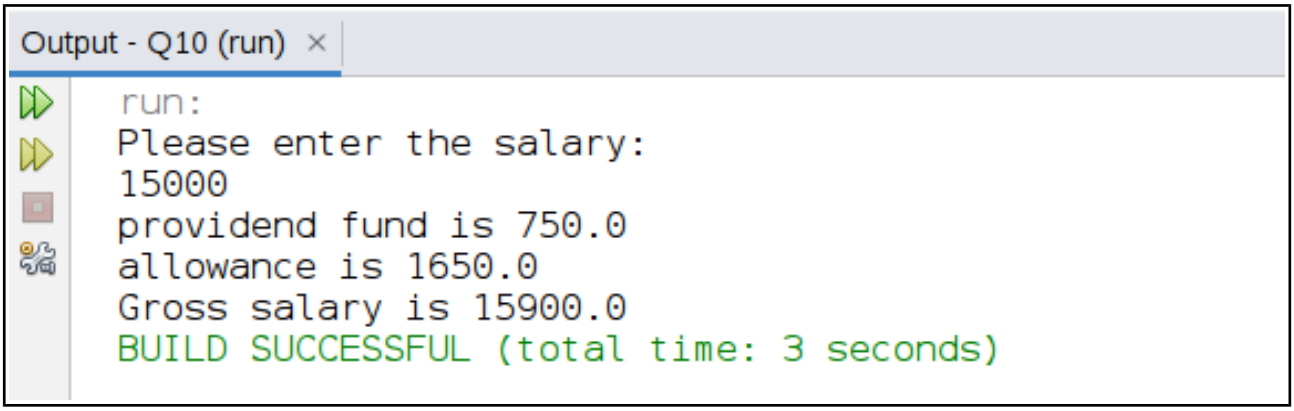
```
package q9;

import java.time.LocalDate;
import java.time.LocalDateTime;
import java.util.Scanner;
import java.util.function.Predicate;

public class Q9 {

    public static void main(String[] args) {
        student obj=new student();
        obj.isEqual();
    }
}
class student{
    public void isEqual()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("please enter first value");
        String s1 =sc.nextLine();
        System.out.println("please enter second value");
        String s2 =sc.nextLine();
        System.out.println(s1.equals(s2));
    }
}
```



**Question 10: Write a program in java with class Employee and do the following operations on it**
**1) Create two constructor default and with Object as parameter to initialize class variables.**
**2) Create a function calculate which calculates the pf and allowances on the salary of employee and return the all values as an object**
**3) Print all the employee an object associated values returned from calculate functions**

---

```
package q10;

import java.util.*;


public class Q10 {


    public static void main(String[] args) {
        Employee obj=new Employee();
        obj.calculate();
    }

}
class Employee{
    public Employee()
    {
    }
    public Employee(int i)
    {
    }
    public void calculate()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Please enter the salary:");
        double salary=sc.nextDouble();
        double pf;
        double al;
        pf=(5*salary)/100;
        al=(11*salary)/100;
        salary=(salary-pf)+al;
        System.out.println("providend fund is "+pf);
        System.out.println("allowance is "+al);
        System.out.println("Gross salary is "+salary);
    }
}
```


Output - Q10 (run)
run:
Please enter the salary:
15000
providend fund is 750.0
allowance is 1650.0
Gross salary is 15900.0
BUILD SUCCESSFUL (total time: 3 seconds)

**Question 11: WAP in java to create Box class with parameterized constructor with an object argument to initialize length, breadth and height also create a function volume which returns the volume of the box and print it in main method.**

---

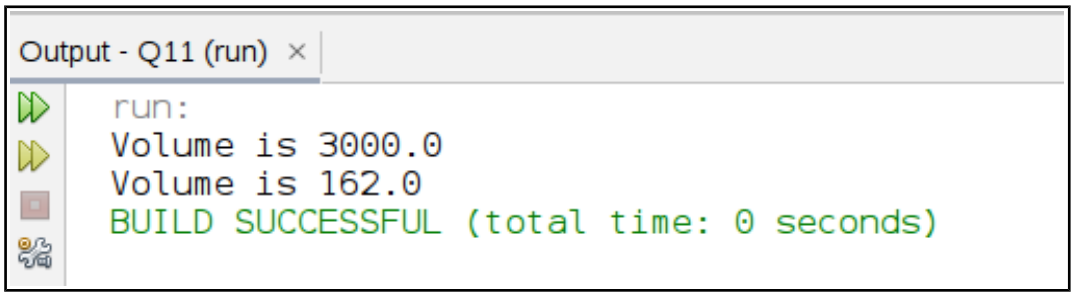
**BOX.JAVA**

```
package pkg11;

/**
 * @author root
 */

class Box {
  double width;
  double height;
  double depth;

  // This is the constructor for Box.
  Box(double w, double h, double d) {
    width = w;
    height = h;
    depth = d;
  }

  double volume() {
    return width * height * depth;
  }
}
```

**BoxDemo.java**

```
package pkg11;

/**
 * @author root
 */
class BoxDemo {
  public static void main(String args[]) {
    Box mybox1 = new Box(10, 20, 15);
    Box mybox2 = new Box(3, 6, 9);

    double vol;

    // get volume of first box
    vol = mybox1.volume();
    System.out.println("Volume is " + vol);
```

```
    // get volume of second box
    vol = mybox2.volume();
    System.out.println("Volume is " + vol);
  }
}
```



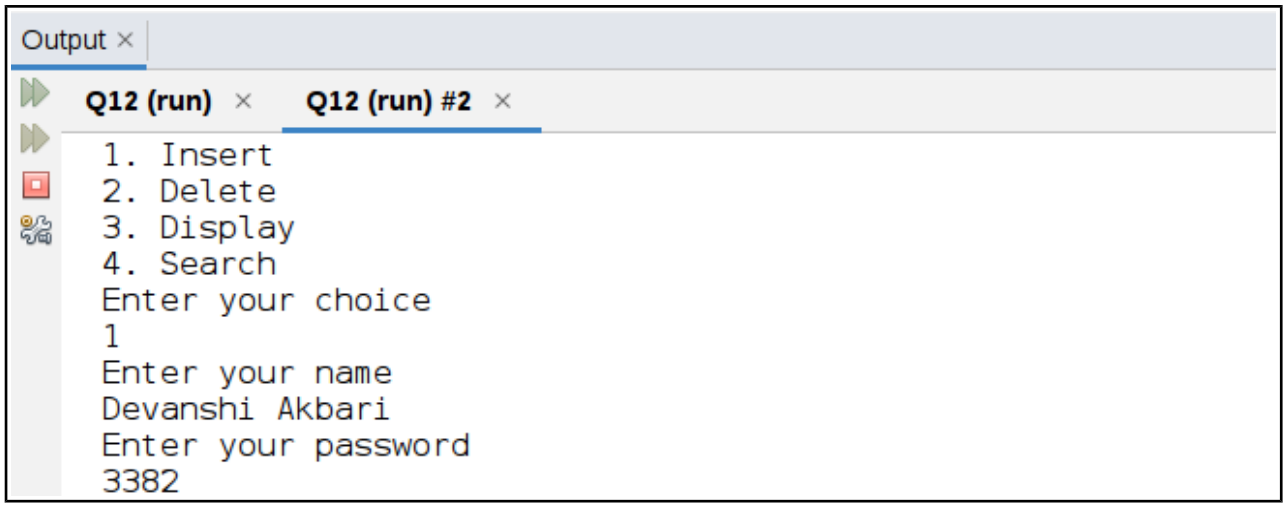
**Question 12: Create a student Record Management system that can perform the following operations:**
**• Insert Student record.**
**• Delete student record**
**• Show student record**
**• Search student record**
**The student record should contain the following items**
**1. Student ID**
**2. Name of Student**
**3. Contact Number of Student**

---

```
package q12;
import java.util.Scanner;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Connection;
import java.sql.PreparedStatement;

public class Q12 {
    static void insert(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your name");
        String n=sc.nextLine();
        System.out.println("Enter your password");
        String c=sc.nextLine();
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student","root","root"
            );
            PreparedStatement ps=con.prepareStatement("insert into stud values(0,'"+n+"','"+c+"')");
            ps.executeUpdate();
            con.close();
            System.out.println("Insertion successfully!!!");
        }catch(Exception e){
             System.out.println("error"+e);
        }
    }
    static void delete(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your id");
        int id=sc.nextInt();
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection
```

```
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student","root","root"

        );
        PreparedStatement ps=con.prepareStatement("delete from stud where id='"+id+"'");
        ps.executeUpdate();
        con.close();
        System.out.println("Deletion successfully!!!");
    }catch(Exception e){
         System.out.println("error"+e);
    }
}
static void display(){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student","root","root" );
        PreparedStatement ps=con.prepareStatement("select * from stud");
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
        System.out.println(rs.getString("id")+"\t"+rs.getString("name"));
        }
        con.close();
    }catch(Exception e){
         System.out.println("error"+e);
    }
}
static void search(){
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter your name");
    String n=sc.nextLine();
    try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection
    cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/student","root","root")
    ;
    PreparedStatement ps=cn.prepareStatement("select * from stud where name='"+n+"'");
    ResultSet rs=ps.executeQuery();
    if(rs.next()){
        System.out.println(rs.getString("id")+"\t"+rs.getString("name"));
    }else{
        System.out.println("no employee");
    }
    }catch(Exception e){
         System.out.println("error"+e);
    }
}
public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int ch=0;
```

```
        while(ch!=5){
        System.out.println("1. Insert");
        System.out.println("2. Delete");
        System.out.println("3. Display");

        System.out.println("4. Search");
        System.out.println("Enter your choice");
        ch=sc.nextInt();
        switch(ch){
            case 1:
            insert();
            break;
            case 2:
            delete();
            break;
            case 3:
            display();
            break;
            case 4:
            search();
            break;
        }
    }
  }
}
```



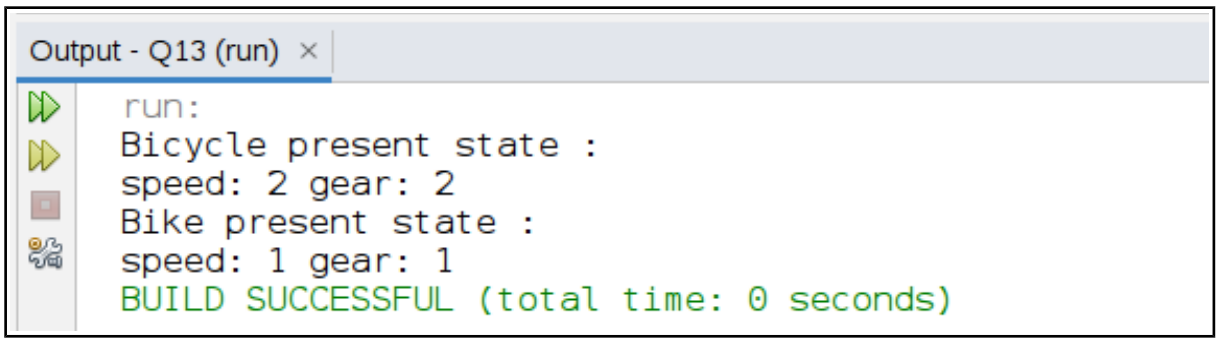
**Question 13: Consider the example of vehicles like bicycle, car, bike....., they have common functionalities. So we make an interface and put all these common functionalities. And lets Bicycle, Bike, car ....etc implement all these functionalities in their own class in their own way.**

---

```
import java.io.*;

interface Vehicle {

	void changeGear(int a);
	void speedUp(int a);
	void applyBrakes(int a);
}

class Bicycle implements Vehicle{

	int speed;
	int gear;

	// to change gear
	@Override
	public void changeGear(int newGear){

		gear = newGear;
	}

	// to increase speed
	@Override
	public void speedUp(int increment){

		speed = speed + increment;
	}

	// to decrease speed
	@Override
	public void applyBrakes(int decrement){

		speed = speed - decrement;
	}

	public void printStates() {
		System.out.println("speed: " + speed
			+ " gear: " + gear);
	}
}

class Bike implements Vehicle {
```

```
	int speed;
	int gear;

	// to change gear
	@Override
	public void changeGear(int newGear){

		gear = newGear;
	}

	// to increase speed
	@Override
	public void speedUp(int increment){

		speed = speed + increment;
	}

	// to decrease speed
	@Override
	public void applyBrakes(int decrement){

		speed = speed - decrement;
	}

	public void printStates() {
		System.out.println("speed: " + speed
			+ " gear: " + gear);
	}

}
class GFG {

	public static void main (String[] args) {

		// creating an inatance of Bicycle

		Bicycle bicycle = new Bicycle();
		bicycle.changeGear(2);
		bicycle.speedUp(3);
		bicycle.applyBrakes(1);

		System.out.println("Bicycle present state :");
		bicycle.printStates();

		// creating instance of bike.
		Bike bike = new Bike();
		bike.changeGear(1);
		bike.speedUp(4);
		bike.applyBrakes(3);
```

```
            System.out.println("Bike present state :");
            bike.printStates();
        }
}
```

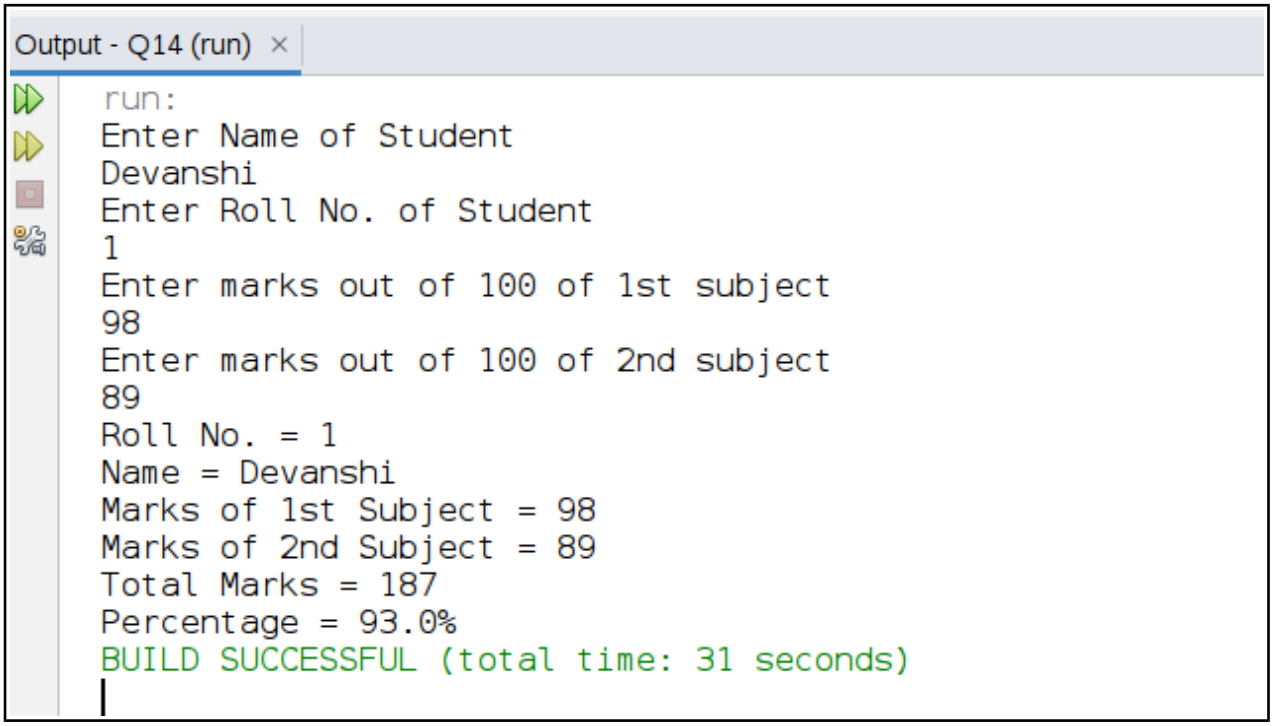


**Question 14: Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating object of Student class (Make use of constructor).**

---

```
import java.io.*;

class Student {

 String name;
 int roll_no;
 int sub1,sub2;

 void getdata() throws IOException {

  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
  System.out.println ("Enter Name of Student");
  name = br.readLine();

  System.out.println ("Enter Roll No. of Student");
  roll_no = Integer.parseInt(br.readLine());

  System.out.println ("Enter marks out of 100 of 1st subject");
  sub1 = Integer.parseInt(br.readLine());

  System.out.println ("Enter marks out of 100 of 2nd subject");
  sub2 = Integer.parseInt(br.readLine());
 }

 void show() {

  int total = sub1+sub2;
  float per = (total * 100) / 200;
  System.out.println ("Roll No. = "+roll_no);
  System.out.println ("Name = "+name);
  System.out.println ("Marks of 1st Subject = "+sub1);
  System.out.println ("Marks of 2nd Subject = "+sub2);
  System.out.println ("Total Marks = "+total);
  System.out.println ("Percentage = "+per+"%");
 }
}

public class StudentDemo {

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

  Student s=new Student();
```

```
 s.getdata();
 s.show();
 }
}
```



**Question15: Write a constructor in the Car class given below that initializes the brand class field with the string "Ford". Call the getBrand() method in the main method of the Sample class and store the value of the brand in a variable, and print the value.**

---

**Car.java**

```
class Car {

	String brand;

	//constructor here
	public Car(){
		this.brand ="Ford";
	}

	public String getBrand() {
		return brand;
	}

	void run() {
		System.out.println("Car is running...");
	}
}
```

**Sample.java**

```
public class Sample {

	public static void main(String[] args) {

				Car ford = new Car();
				String brand = ford.getBrand();
				System.out.println(brand);
	}
}
```