JAVA OOPs CONCEPTS

Abstraction:

abstract class Bank{

abstract int getInterestRate();

}

class indianbank extends Bank{

int getInterestRate()

{

return 5;

}

}

class SBI extends Bank{

int getInterestRate()

{

return 4;

}

}

class Main{

public static void main(String args[]){

Bank b;

b = new indianbank ();

System.out.println("Indian Bank Rate of Interest is: "+b.getInterestRate()+"%");

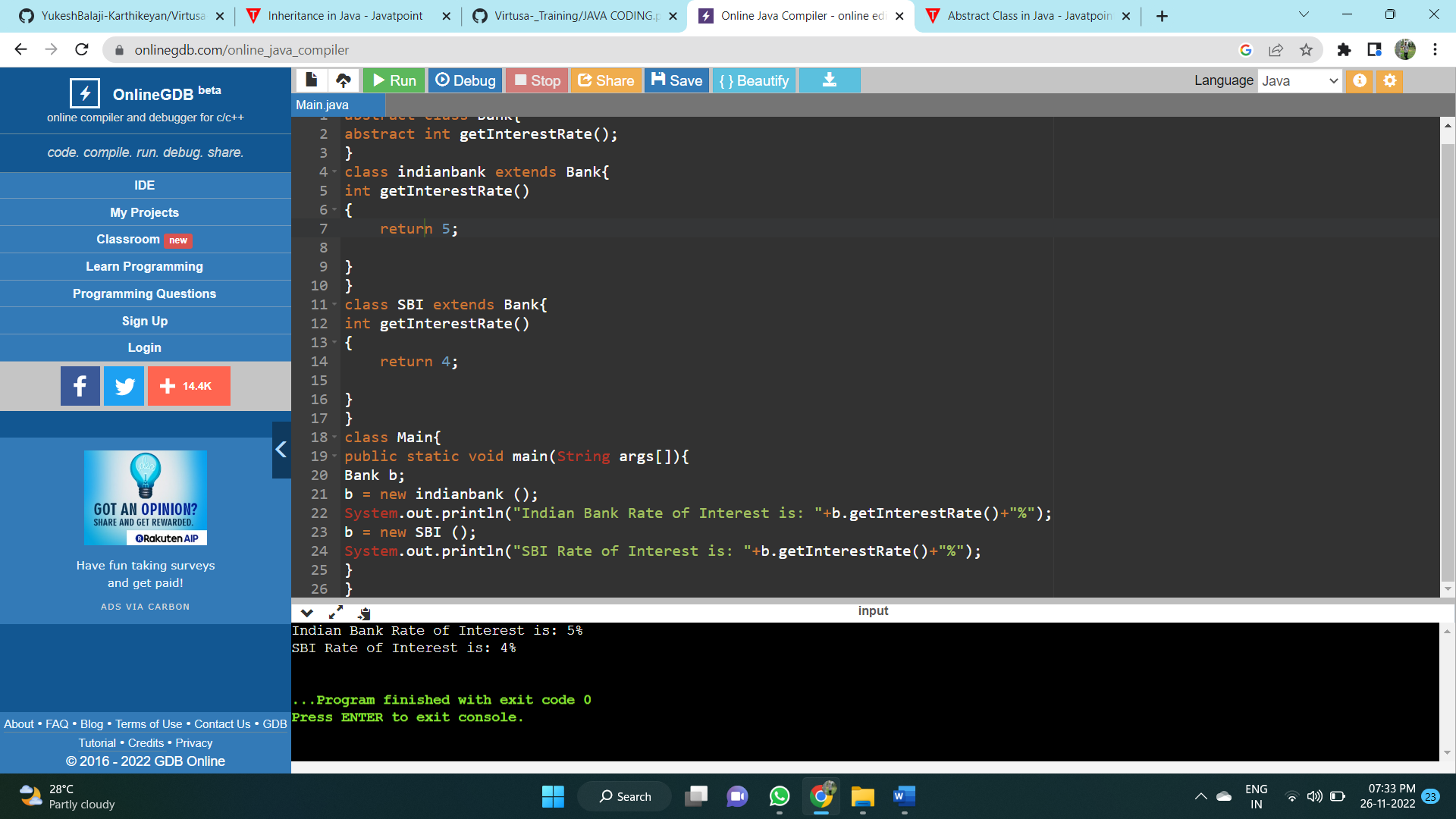
b = new SBI ();

System.out.println("SBI Rate of Interest is: "+b.getInterestRate()+"%");

}

}

Screenshot of the Output:



Encapsulation:

class Employee {

private int Emp\_Id;

private String name;

public int getId() {

return Emp\_Id;

}

public void setId(int s\_id) {

this.Emp\_Id = s\_id;

}

public String getname() {

return name;

}

public void setname(String s\_name) {

this.name = s\_name;

}

}

class Main{

public static void main(String[] args) {

Employee e=new Employee();

e.setId (50);

e.setname("Yukesh Balaji");

System.out.println("Employee Details:" + "\nEmployee ID:" + e.getId()

+ " \nEmployee Name:" + e.getname());

}

}

Screenshot of the Output:

