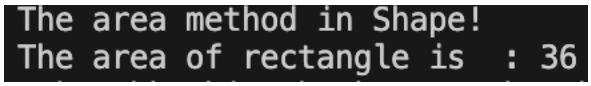
Question 01 : Write a Java program to create a class called Shape with a method called getArea().Create a subclass called Rectangle that overrides the getArea() method to calculate the area of a rectangle.

Code:

```
class Shape
{
public void getArea()
{
System.out.println("The area method in Shape!");
}
}

class Rectangle extends Shape
{
public int len, bre;
Rectangle()
{

}
Rectangle (int len, int bre)
{
this.len = len;
this.bre = bre;
}
@Override
public void getArea()
```

```
{
System.out.println("The area of rectangle is : "+(len*bre));
}
}

public class aGetShapeAreaOverride{
public static void main(String[] args) {
Shape ref = new Shape();
ref.getArea();
ref=new Rectangle(12,3) ;
ref. getArea();
}
}
```

Output :



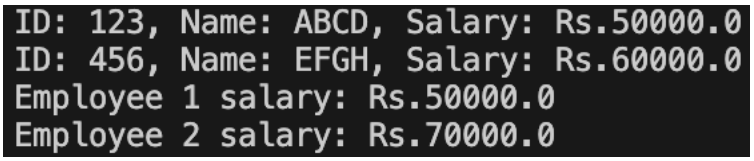
Question 02 : Write a Java program to create a class called Employee with methods called showEmployee() and getSalary(). Create a subclass called HRManager that overrides the getSalary() (employee, id, name, salary, hr)

Code:

```
class Employee {
private int id;
private String name;
private double salary;

public Employee(int id, String name, double salary) {
this.id = id;
this.name = name;
this.salary = salary;
}

public void showEmployee() {
System.out.println("ID: " + id + ", Name: " + name + ", Salary: Rs." + salary);
}

public double getSalary() {
return salary;
}
}



class HRManager extends Employee {
private double bonus;
```

```
public HRManager(int id, String name, double salary, double bonus) {
super(id, name, salary);
this.bonus = bonus;
}

@Override
public double getSalary() {
return super.getSalary() + bonus;
}
}

public class bEmployeeSalOverride {
public static void main(String[] args) {
Employee emp1 = new Employee(123, "ABCD", 50000);
HRManager emp2 = new HRManager(456, "EFGH", 60000, 10000);

emp1.showEmployee();
emp2.showEmployee();

System.out.println("Employee 1 salary: Rs." + emp1.getSalary());
System.out.println("Employee 2 salary: Rs." + emp2.getSalary());
}
}
```

Output:


ID: 123, Name: ABCD, Salary: Rs.50000.0
ID: 456, Name: EFGH, Salary: Rs.60000.0
Employee 1 salary: Rs.50000.0
Employee 2 salary: Rs.70000.0

Question03: Write a Java program to create a class known as "BankAccount" with methods called deposit() and withdraw(). Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

Code:

```
class BankAccount {
protected double balance;

public BankAccount(double initialBalance) {
this.balance = initialBalance;
}

public void deposit(double amount) {
if (amount > 0) {
System.out.print("\nDeposit successful of "+amount+" Prev Balance: Rs."+balance);
balance += amount;
System.out.print(". New balance: Rs." + balance +"\n");
} else {
System.out.println("\nDeposit amount must be positive.");
}
}

public void withdraw(double amount) {
if (amount > 0 && balance >= amount) {
System.out.print("\nWithdrawal successful of Rs."+amount+" Prev balance: Rs." + balance);
balance -= amount;
System.out.print("New balance: Rs." + balance+"\n");
```

```
} else if (amount <= 0) {
System.out.println("\nWithdrawal amount must be positive.");
} else {
System.out.println("\nInsufficient funds. Withdrawal failed.");
}
}
}

class SavingsAccount extends BankAccount {
private final double minimumBalance = 1000;

public SavingsAccount(double initialBalance) {
super(initialBalance);
if (initialBalance < minimumBalance) {
System.out.println("\nSavings account requires a minimum balance of Rs." + minimumBalance);
}
}

@Override
public void withdraw(double amount) {
if (amount > 0 && balance - amount >= minimumBalance) {
super.withdraw(amount);
} else {
System.out.println("\nWithdrawal failed. Minimum balance of Rs." + minimumBalance + " must
be maintained.");
}
}
}
```

```
public class cBankSystemOverride {
public static void main(String[] args) {
BankAccount account1 = new BankAccount(2000);
SavingsAccount account2 = new SavingsAccount(2000);

account1.deposit(1000);
account1.withdraw(500);

account2.deposit(1000);
account2.withdraw(2100);
account2.withdraw(2000);
}
}
```

Output:

```
Deposit successful of 1000.0 Prev Balance: Rs.2000.0. New balance: Rs.3000.0
Withdrawal successful of Rs.500.0 Prev balance: Rs.3000.0New balance: Rs.2500.0
Deposit successful of 1000.0 Prev Balance: Rs.2000.0. New balance: Rs.3000.0
Withdrawal failed. Minimum balance of Rs.1000.0 must be maintained.
Withdrawal successful of Rs.2000.0 Prev balance: Rs.3000.0New balance: Rs.1000.0
```