**EXERCISE -** 5

**CONSTRUCTORS & OVERLOADING**

**Aim:** Write a java program to implement Default Constructor

**File name:** DefConstructor.java

**Program:**

**//Program on Default constructor**

**import java.lang.\*;**

**class Student{**

**String Name;**

**String RollNo;**

**String Branch;**

**double percentage;**

**void GetDetails(){**

**System.out.println("Name: "+Name+"\nRollNo: "+RollNo+"\nBranch:"+ Branch+"\nPercentage:"+percentage);**

**}**

**}**

**class DefConstructor{**

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

**Student srinu = new Student(); //Default Constructor**

**srinu.Name = "Srinivas Rao T";**

**srinu.RollNo = "217Y1A05C0";**

**srinu.Branch = "CSE";**

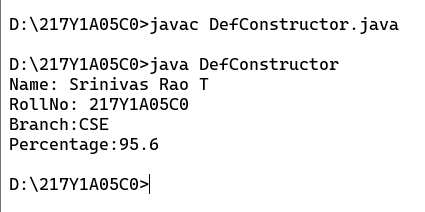
**srinu.percentage = 95.6;**

**srinu.GetDetails();**

**}**

**}**

**Output:**

****

**Aim:** Write a java program to implement Non Argument Constructor

**File name:** NArgConstructor.java

**Program:**

**//Program on Non Argument Constructor**

**import java.lang.\*;**

**class Student{**

**String Name;**

**String RollNo;**

**String Branch;**

**double percentage;**

**Student(){**

**Name = "Srinivas Rao T";**

**RollNo = "217Y1A05C0";**

**Branch = "CSE";**

**percentage = 95.6;**

**}**

**void GetDetails(){**

**System.out.println("Name: "+Name+"\nRollNo: "+RollNo+"\nBranch:"+ Branch+"\nPercentage:"+percentage);**

**}**

**}**

**class NArgConstructor{**

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

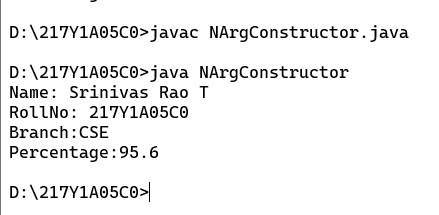
**Student srinu = new Student(); //Non Argument Constructor**

**srinu.GetDetails();**

**}**

**}**

**Output:**

****

**Aim:** Write a java program to implement Parameterised Constructor

**File name:** PArConstructor.java

**Program:**

**//Program on Parameterised Constructor**

**import java.lang.\*;**

**class Student{**

**String Name;**

**String RollNo;**

**String Branch;**

**double percentage;**

**Student(String Name,String RollNo,String Branch,double percentage){**

**this.Name = Name;**

**this.RollNo = RollNo;**

**this.Branch = Branch;**

**this.percentage = percentage;**

**}**

**void GetDetails(){**

**System.out.println("Name: "+Name+"\nRollNo: "+RollNo+"\nBranch: "+ Branch+"\nPercentage: "+percentage);**

**}**

**}**

**class PArConstructor{**

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

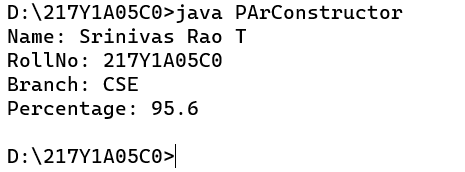
**Student srinu = new Student("Srinivas Rao T","217Y1A05C0","CSE",95.6); //Parameterised Constructor**

**srinu.GetDetails();**

**}**

**}**

**Output:**

****

**Aim:** Write a java program to implement Constructor overloading

**File name:** ConOverload.java

**Program:**

**//Program on Constructor overloading**

**import java.lang.\*;**

**class Student{**

**String Name;**

**String RollNo;**

**String Branch;**

**double percentage;**

**Student(){**

**Name = "Srinivas Rao T";**

**RollNo = "217Y1A05C0";**

**Branch = "CSE";**

**percentage = 95.6;**

**}**

**Student(String Name,String RollNo,String Branch,double percentage){**

**this.Name = Name;**

**this.RollNo = RollNo;**

**this.Branch = Branch;**

**this.percentage = percentage;**

**}**

**void GetDetails(){**

**System.out.println("Name: "+Name+"\nRollNo: "+RollNo+"\nBranch: "+ Branch+"\nPercentage: "+percentage);**

**}**

**}**

**class ConOverload{**

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

**Student ravi = new Student();**

**Student srinu = new Student("Sriraj D","217Y1A05C1","CSE",98.4); //Parameterised Constructor**

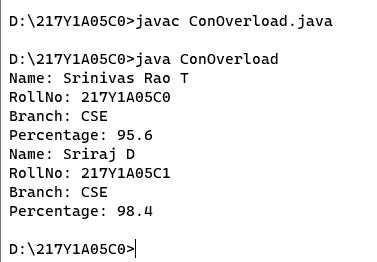
**ravi.GetDetails();**

**srinu.GetDetails();**

**}**

**}**

**Output:**

****

**Aim:** Write a java program to implement Method overloading

**File name:** MetOverload.java

**Program:**

**//Program on Method overloading**

**import java.lang.\*;**

**class Addition{**

**void add(){**

**System.out.println(0);**

**}**

**void add(int a,int b,int c){**

**System.out.println((a+b+c));**

**}**

**void add(int a,int b){**

**System.out.println((a+b));**

**}**

**}**

**class MetOverload{**

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

**Addition a = new Addition();**

**a.add();**

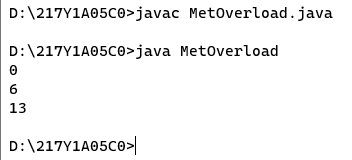
**a.add(1,2,3);**

**a.add(6,7);**

**}**

**}**

**Output:**

****