MODUL 5

MEMBUAT CLASS SENDIRI

A. TUJUAN

Setelah praktikum ini, praktikan diharapkan dapat memahami dan membuat class sendiri, *overloading method*, dan *overloading constructor* dengan bahasa pemrograman java.

B. PERALATAN DAN BAHAN

1. Personal Komputer
2. Software *Netbeans*

C. TEORI

- ***Overloading Method***

  Overloading method merupakan sebuah kondisi dimana sebuah class memiliki 2 atau lebih method dengan nama yang sama, namun memiliki jumlah parameter yang berbeda, tipe paramater dan urutan dari tipe data paramater yang berbeda.

- ***Overloading Constructor***

  Overloading constructor merupakan sebuah kondisi dimana terdapat konstruktor yang lebih dari satu dalam sebuah Class, namun memiliki parameter yang berbeda, jumlah parameternya yang berbeda, dan type data parameter yang berbeda.

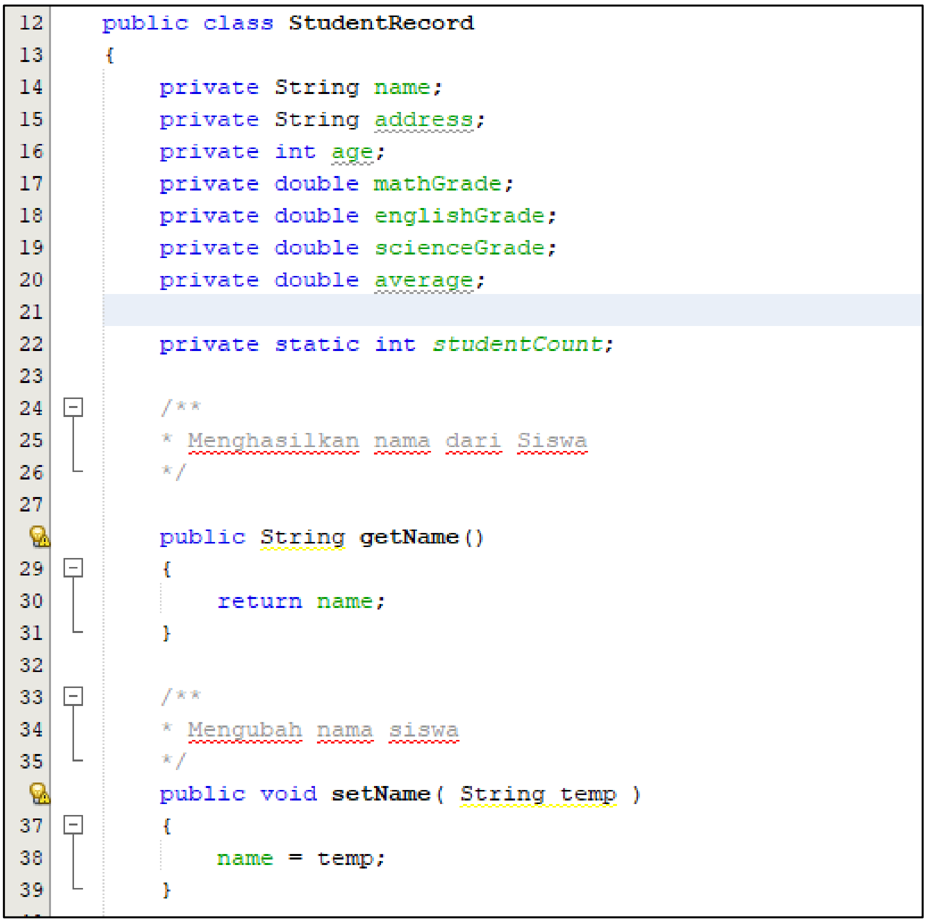
- **Keyword this**

  Keyword this merupakan keyword yang berfungsi untuk mereferensikan atau mengacu ke objek yang sedang aktif. Keyword this sering digunakan pada overload method dan overload constructor.

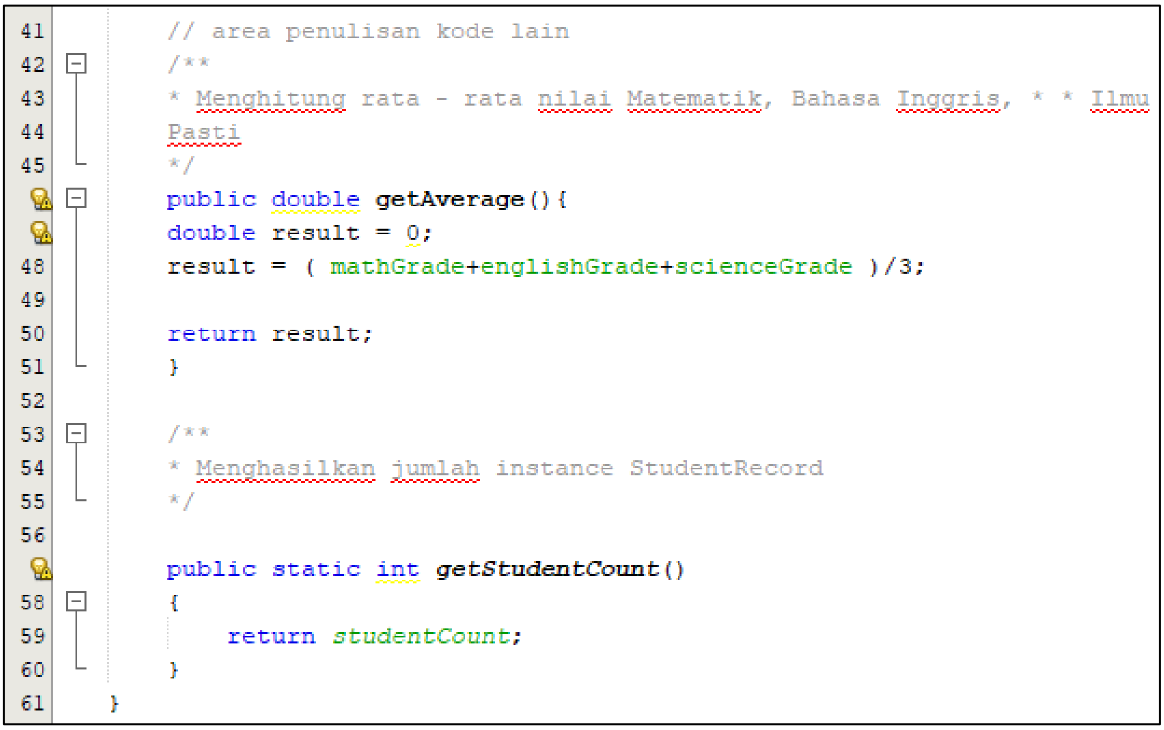
D. PRAKTIKUM

1. Membuat Class sendiri :

❖ **<u>Code Class StudentRecord :</u>**

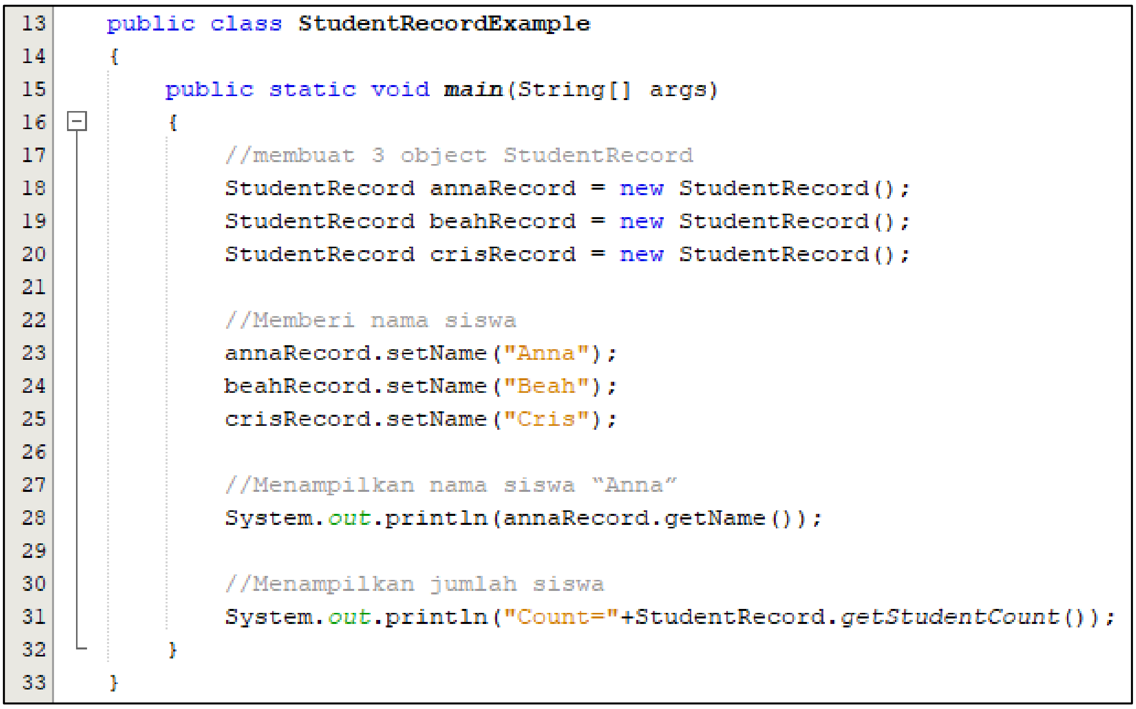


```
public class StudentRecord
{
    private String name;
    private String address;
    private int age;
    private double mathGrade;
    private double englishGrade;
    private double scienceGrade;
    private double average;

    private static int studentCount;

    /**
    * Menghasilkan nama dari Siswa
    */

    public String getName()
    {
        return name;
    }

    /**
    * Mengubah nama siswa
    */
    public void setName( String temp )
    {
        name = temp;
    }
```



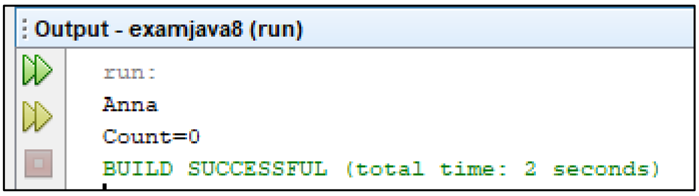
```
    // area penulisan kode lain
    /**
    * Menghitung rata - rata nilai Matematik, Bahasa Inggris, * * Ilmu
    Pasti
    */
    public double getAverage(){
    double result = 0;
    result = ( mathGrade+englishGrade+scienceGrade )/3;

    return result;
    }

    /**
    * Menghasilkan jumlah instance StudentRecord
    */

    public static int getStudentCount()
    {
        return studentCount;
    }
}
```

- ❖ **Code StudentRecordExample :**



```java
public class StudentRecordExample
{
    public static void main(String[] args)
    {
        //membuat 3 object StudentRecord
        StudentRecord annaRecord = new StudentRecord();
        StudentRecord beahRecord = new StudentRecord();
        StudentRecord crisRecord = new StudentRecord();

        //Memberi nama siswa
        annaRecord.setName("Anna");
        beahRecord.setName("Beah");
        crisRecord.setName("Cris");

        //Menampilkan nama siswa "Anna"
        System.out.println(annaRecord.getName());

        //Menampilkan jumlah siswa
        System.out.println("Count="+StudentRecord.getStudentCount());
    }
}
```

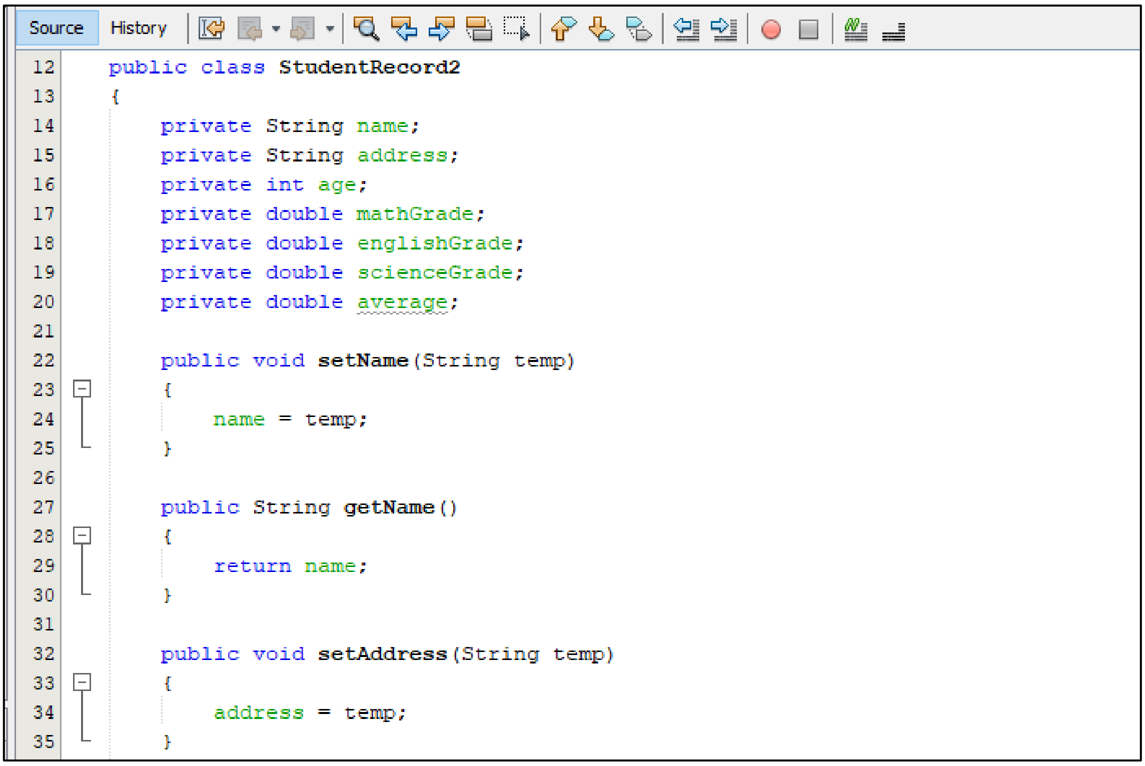
- ❖ ***Output :***



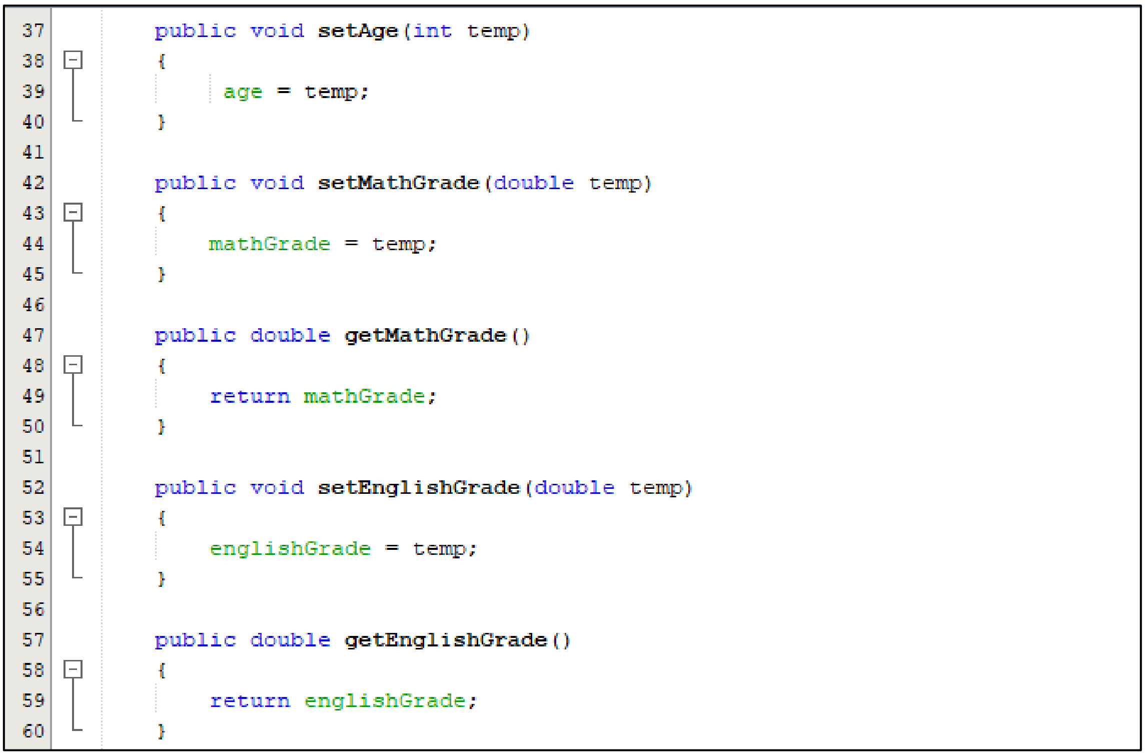
```
run:
Anna
Count=0
BUILD SUCCESSFUL (total time: 2 seconds)
```

2. *Overloading Method*

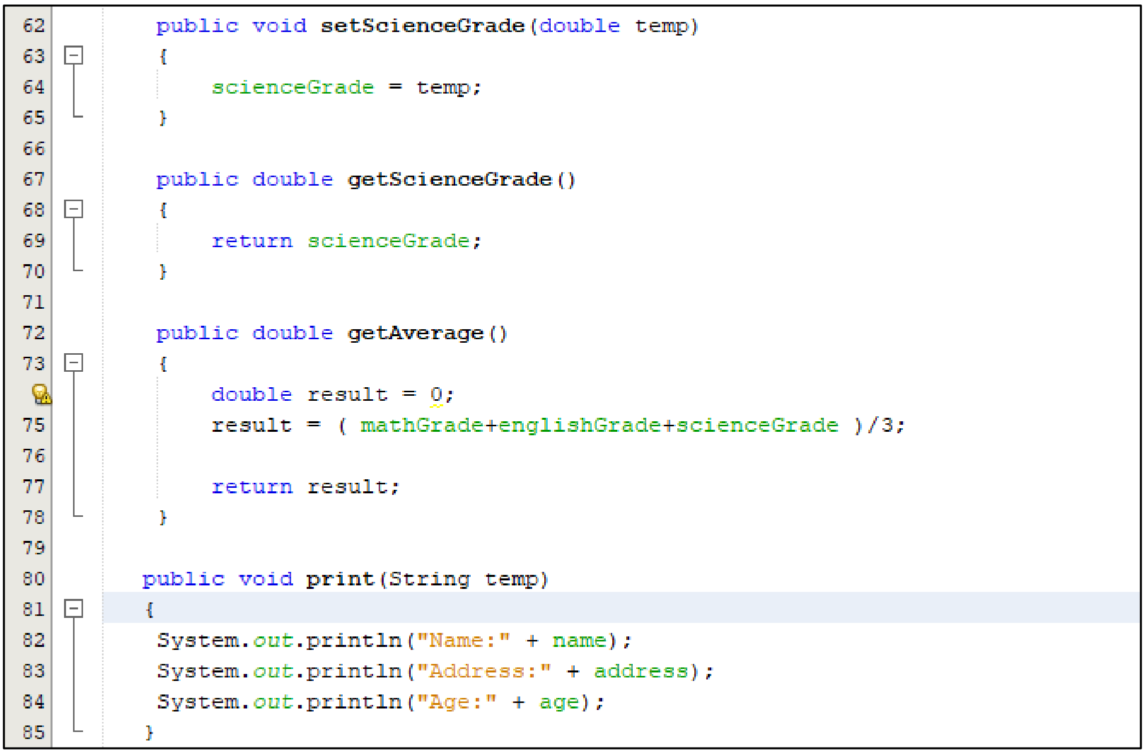
- ❖ **Code :**



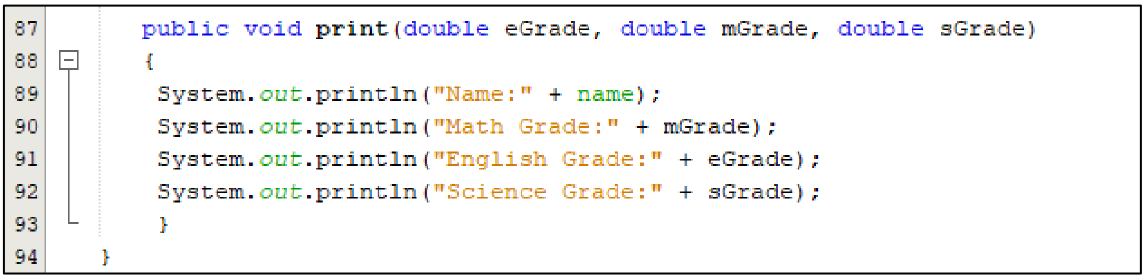
```java
public class StudentRecord2
{
    private String name;
    private String address;
    private int age;
    private double mathGrade;
    private double englishGrade;
    private double scienceGrade;
    private double average;

    public void setName(String temp)
    {
        name = temp;
    }

    public String getName()
    {
        return name;
    }

    public void setAddress(String temp)
    {
        address = temp;
    }
```



```java
    public void setAge(int temp)
    {
        age = temp;
    }

    public void setMathGrade(double temp)
    {
        mathGrade = temp;
    }

    public double getMathGrade()
    {
        return mathGrade;
    }

    public void setEnglishGrade(double temp)
    {
        englishGrade = temp;
    }

    public double getEnglishGrade()
    {
        return englishGrade;
    }
```

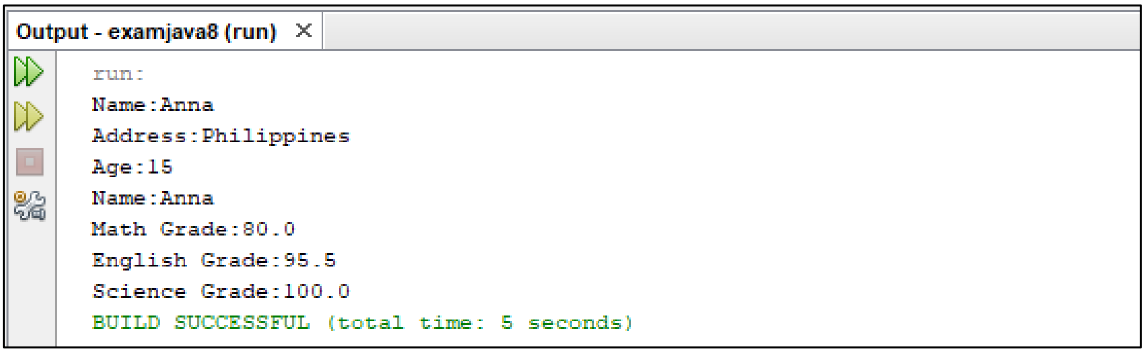


```java
     public void setScienceGrade(double temp)
     {
         scienceGrade = temp;
     }

     public double getScienceGrade()
     {
         return scienceGrade;
     }

     public double getAverage()
     {
         double result = 0;
         result = ( mathGrade+englishGrade+scienceGrade )/3;

         return result;
     }

    public void print(String temp)
    {
     System.out.println("Name:" + name);
     System.out.println("Address:" + address);
     System.out.println("Age:" + age);
    }
```



```java
    public void print(double eGrade, double mGrade, double sGrade)
    {
     System.out.println("Name:" + name);
     System.out.println("Math Grade:" + mGrade);
     System.out.println("English Grade:" + eGrade);
     System.out.println("Science Grade:" + sGrade);
     }
 }
```

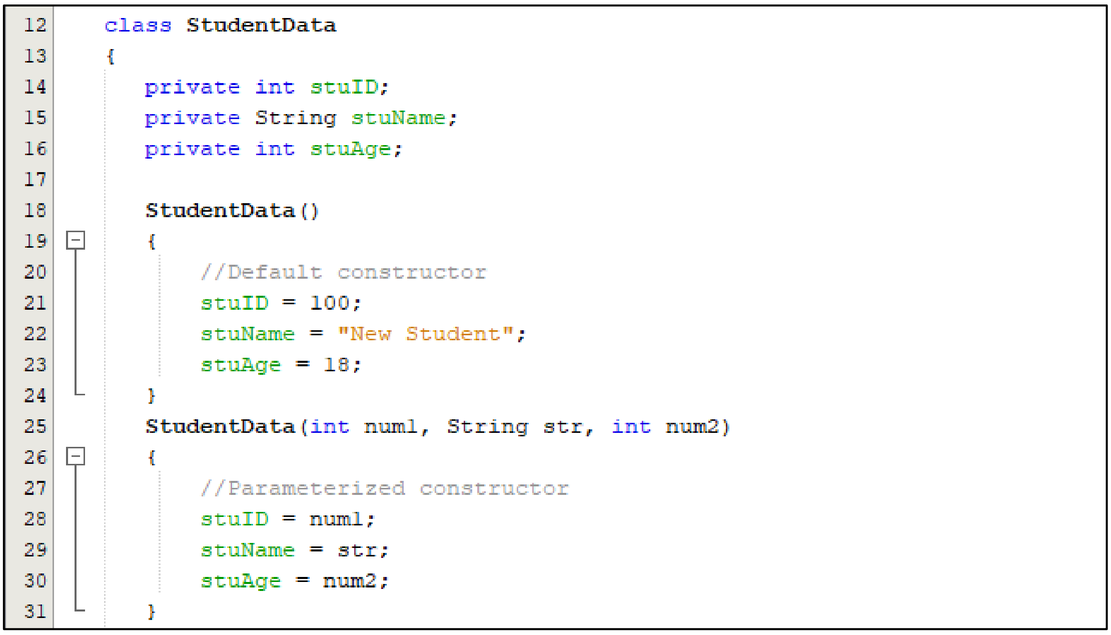
❖ ***Output :***



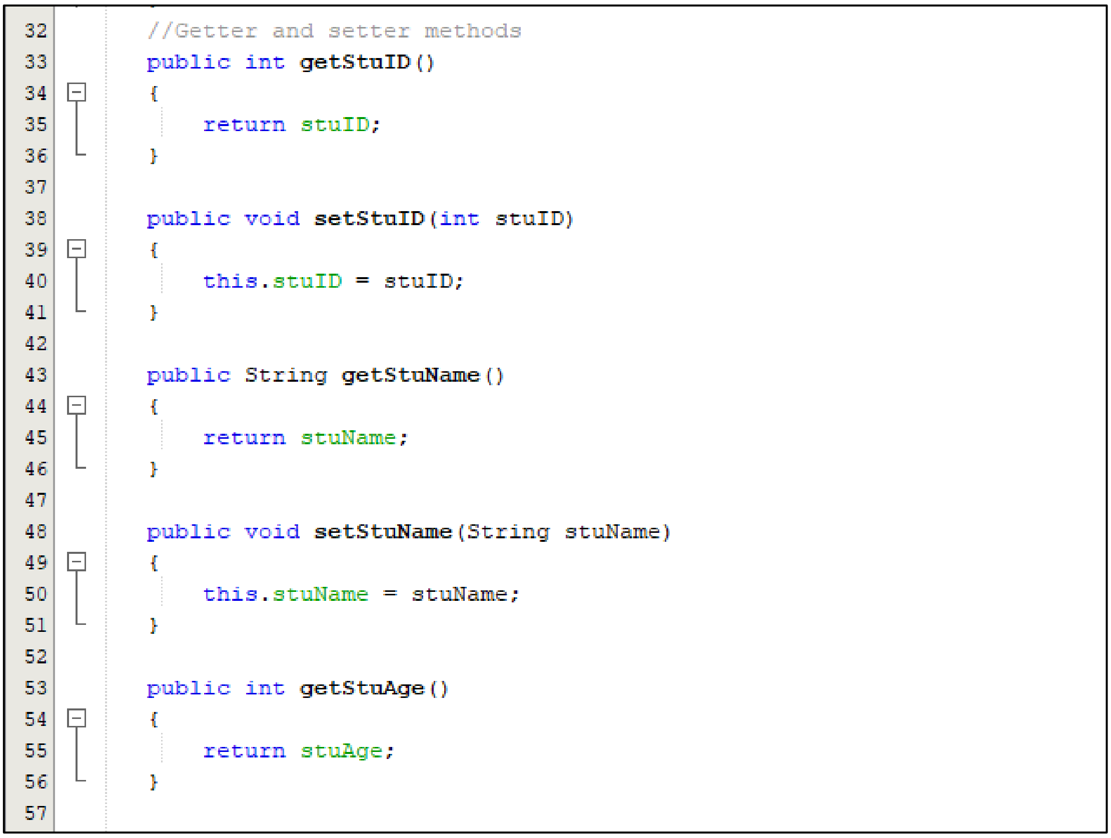
```
run:
Name:Anna
Address:Philippines
Age:15
Name:Anna
Math Grade:80.0
English Grade:95.5
Science Grade:100.0
BUILD SUCCESSFUL (total time: 5 seconds)
```

3. Overloading Constructor :

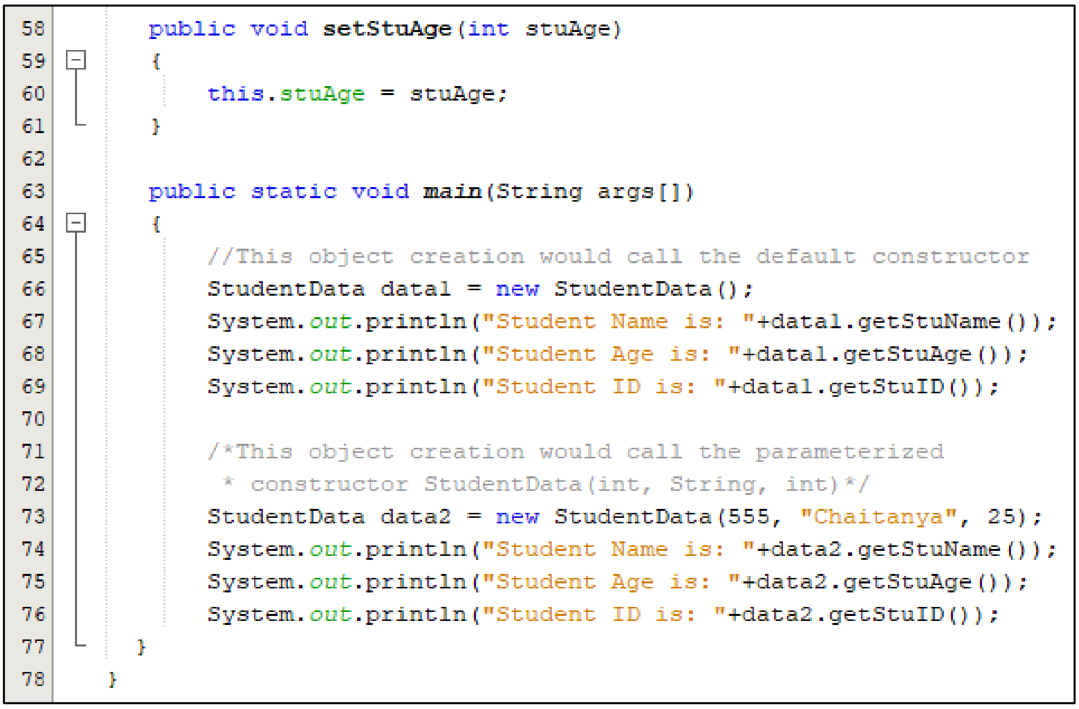
❖ Code :



```
class StudentData
{
    private int stuID;
    private String stuName;
    private int stuAge;

    StudentData()
    {
        //Default constructor
        stuID = 100;
        stuName = "New Student";
        stuAge = 18;
    }
    StudentData(int num1, String str, int num2)
    {
        //Parameterized constructor
        stuID = num1;
        stuName = str;
        stuAge = num2;
    }
```

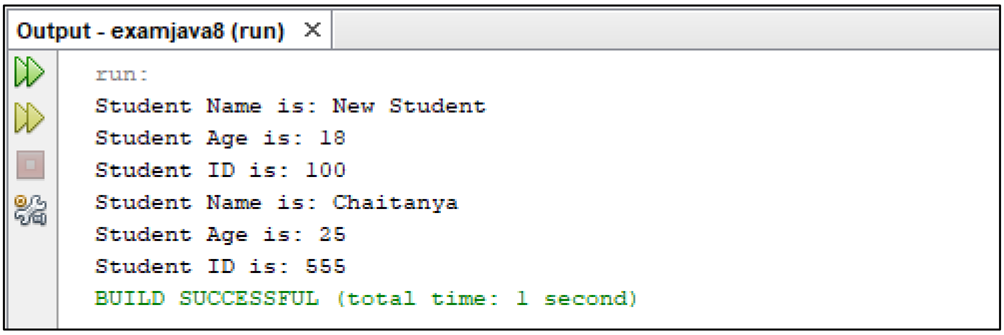


```
    //Getter and setter methods
    public int getStuID()
    {
        return stuID;
    }

    public void setStuID(int stuID)
    {
        this.stuID = stuID;
    }

    public String getStuName()
    {
        return stuName;
    }

    public void setStuName(String stuName)
    {
        this.stuName = stuName;
    }

    public int getStuAge()
    {
        return stuAge;
    }

```



```java
    public void setStuAge(int stuAge)
    {
        this.stuAge = stuAge;
    }

    public static void main(String args[])
    {
        //This object creation would call the default constructor
        StudentData data1 = new StudentData();
        System.out.println("Student Name is: "+data1.getStuName());
        System.out.println("Student Age is: "+data1.getStuAge());
        System.out.println("Student ID is: "+data1.getStuID());

        /*This object creation would call the parameterized
         * constructor StudentData(int, String, int)*/
        StudentData data2 = new StudentData(555, "Chaitanya", 25);
        System.out.println("Student Name is: "+data2.getStuName());
        System.out.println("Student Age is: "+data2.getStuAge());
        System.out.println("Student ID is: "+data2.getStuID());
    }
}
```

❖ **Output :**



```
run:
Student Name is: New Student
Student Age is: 18
Student ID is: 100
Student Name is: Chaitanya
Student Age is: 25
Student ID is: 555
BUILD SUCCESSFUL (total time: 1 second)
```

E. TUGAS

1. Buatlah program *overloading method* yang dapat menampilkan identitas mahasiswa dan prestasi akademiknya, seperti gambaran berikut :

Identitas Mahasiswa	Prestasi Akademik
Nama	Nama
Nim	IP Semester
Jurusan	IP Kumulatif
Fakultas	Riwayat Beasiswa

2. Buatlah program *overloading constructor* yang dapat menampilkan identitas mahasiswa seperti gambaran berikut :

Identitas Mahasiswa
Nama
Nim
Jurusan
Fakultas
IP Semester
IP Kumulatif