Chapter 15

Creating Objects

OBJECT → CL

```java
class Student{
    String name;
    String studyClass;
    int rollno;
    double percentage;

    void setStudyClass(String sc){
        studyClass = sc;
    }

    void setName(String n){
        name = n;
    }

    void setRollno(int r){
        rollno = r;
    }
    void setPercentage(double p){
        percentage = p;
    }

    String getName(){
        return name;
    }

    String getStudyClass(){
        return studyClass;
    }

    int getRollno(){
        return rollno;
    }
    double getPercentage(){
        return percentage;
    }

}
```

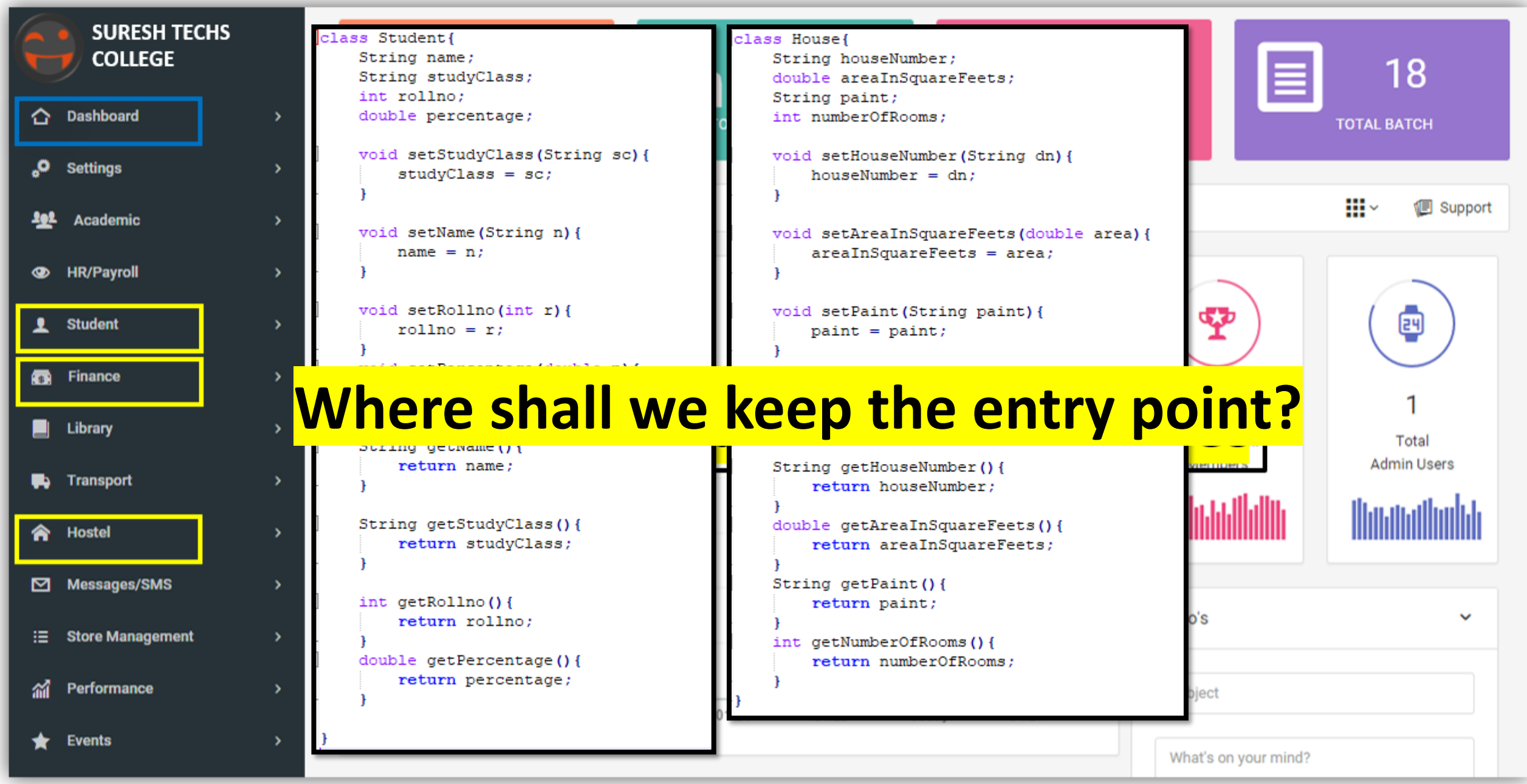
```java
class House{
    String houseNumber;
    double areaInSquareFeets;
    String paint;
    int numberOfRooms;

    void setHouseNumber(String dn){
        houseNumber = dn;
    }

    void setAreaInSquareFeets(double area){
        areaInSquareFeets = area;
    }

    void setPaint(String paint){
        paint = paint;
    }

    void setNumberOfRooms(int nOfRooms){
        numberOfRooms = nOfRooms;
    }

    String getHouseNumber(){
        return houseNumber;
    }
    double getAreaInSquareFeets(){
        return areaInSquareFeets;
    }
    String getPaint(){
        return paint;
    }
    int getNumberOfRooms(){
        return numberOfRooms;
    }
}
```

OBJECT IN PROGRAM

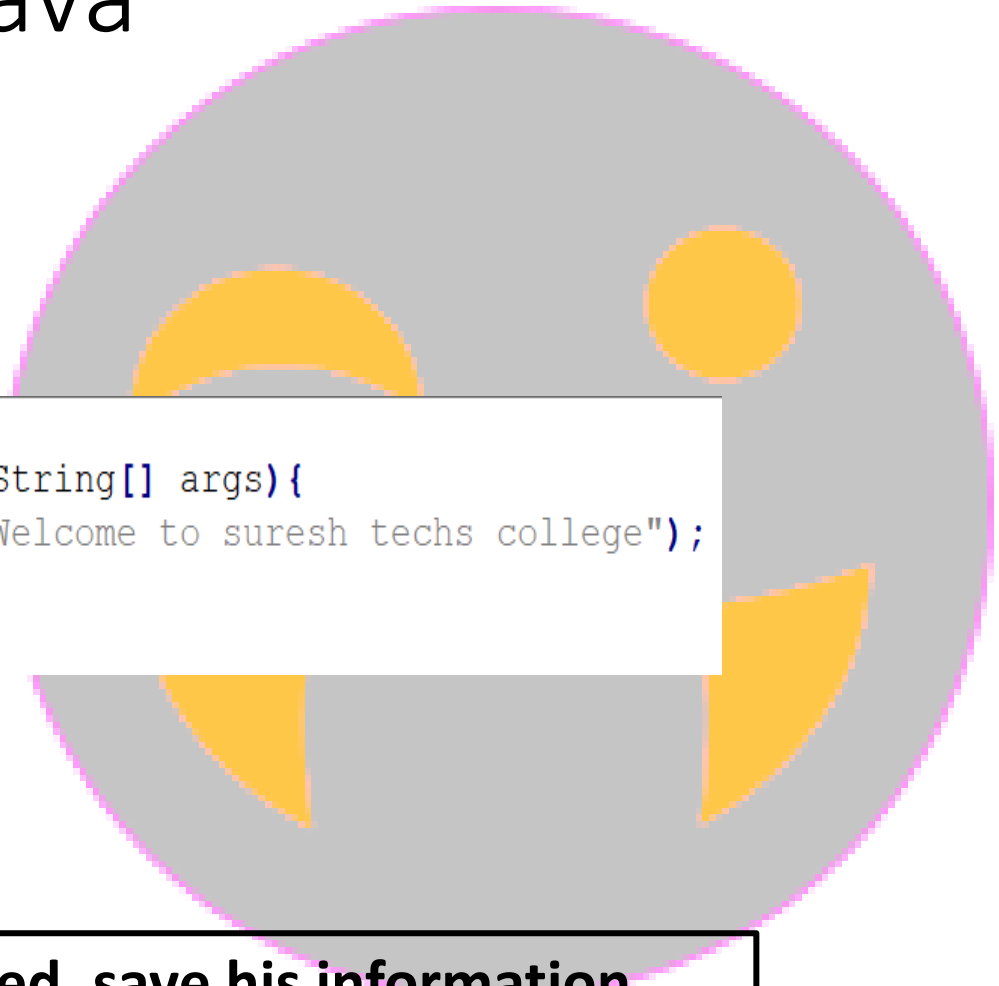
Student s = new Student()

Application entry point

Every application will have an entry point(main gate) from where execution starts


SURESH TECHS COLLEGE
Dashboard
Settings
Academic
HR/Payroll
Student
Finance
Library
Transport
Hostel
Messages/SMS
Store Management
Performance
Events
class Student{
String name;
String studyClass;
int rollno;
double percentage;
void setStudyClass(String sc){
studyClass = sc;
}
void setName(String n){
name = n;
}
void setRollno(int r){
rollno = r;
}
return name;
}
String getStudyClass(){
return studyClass;
}
int getRollno(){
return rollno;
}
double getPercentage(){
return percentage;
}
}
class House{
String houseNumber;
double areaInSquareFeets;
String paint;
int numberOfRooms;
void setHouseNumber(String dn){
houseNumber = dn;
}
void setAreaInSquareFeets(double area){
areaInSquareFeets = area;
}
void setPaint(String paint){
paint = paint;
}
String getHouseNumber(){
return houseNumber;
}
double getAreaInSquareFeets(){
return areaInSquareFeets;
}
String getPaint(){
return paint;
}
int getNumberOfRooms(){
return numberOfRooms;
}
}
Where shall we keep the entry point?
18
TOTAL BATCH
Support
1
Total
Admin Users
What's on your mind?

Dashboard.java



```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
    }
}
```

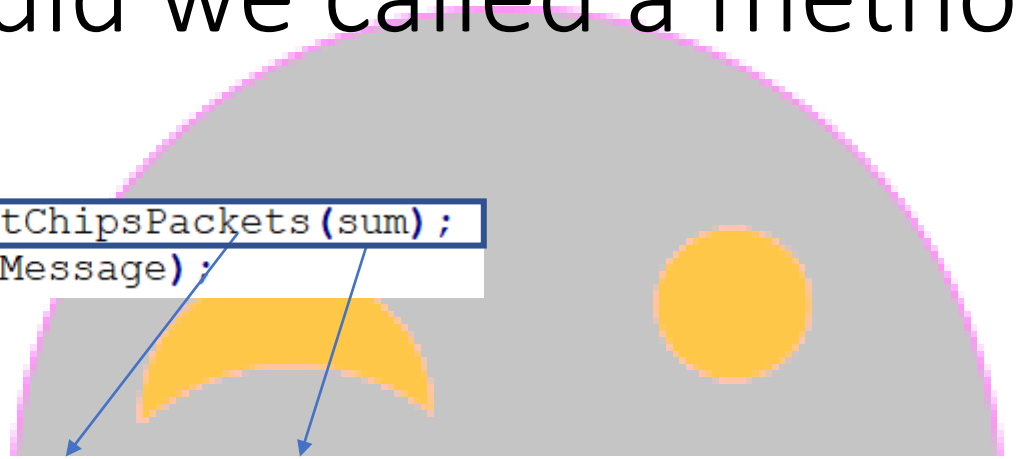
First student joined, save his information

```java
class Student{
    String name;
    String studyClass;
    int rollno;
    double percentage;

    void setStudyClass(String sc){
        studyClass = sc;
    }

    void setName(String n){
        name = n;
    }

    void setRollno(int r){
        rollno = r;
    }
    void setPercentage(double p){
        percentage = p;
    }

    String getName(){
        return name;
    }

    String getStudyClass(){
        return studyClass;
    }

    int getRollno(){
        return rollno;
    }
    double getPercentage(){
        return percentage;
    }

}
```

Recall, how did we called a method?

int sum = 20;

```
String chipsMessage = getChipsPackets(sum);
System.out.println(chipsMessage);
```



```
static String getChipsPackets(int cash){
    if(cash<10){
        return "Sorry, minimum Rs. 10";
    }else{
        int chipsPackets = cash/10;
        int reminingAmount = cash%10;
        String message = "e "+chipsPackets+" chips packets teesukondi.";
        if(reminingAmount!=0){
            message = message + "e Rs."+reminingAmount+" chillara teesukondi";
        }
        return message;
    }
}
```

2

2

Let’s represent a student

```
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        new Student();
    }
}
```

When ever an object is created, constructor will be called

Constructor will not have a return type

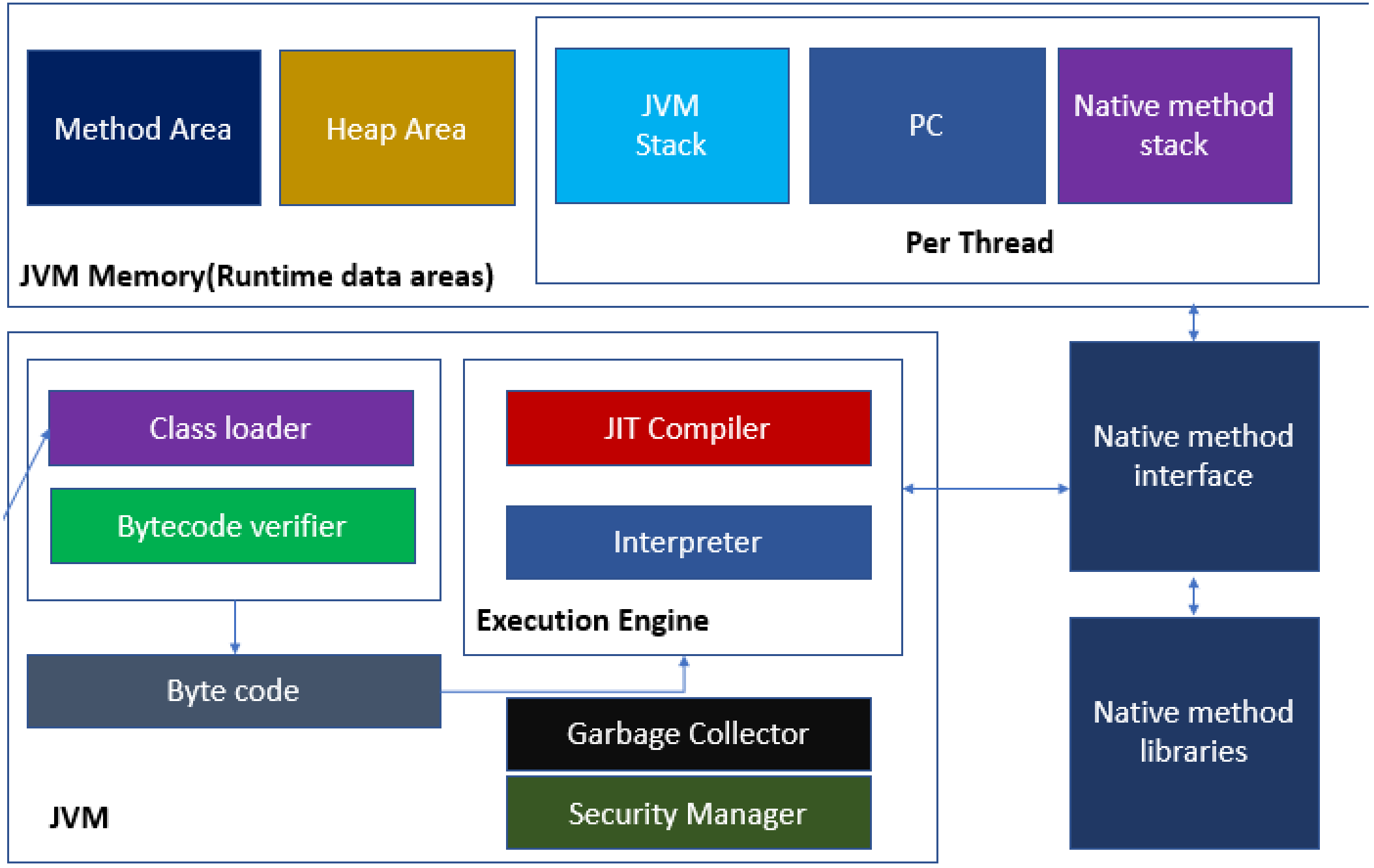
Can be used to construct default state for an object

```
class Student{
    String name;
    String studyClass;
    int rollno;
    double percentage;

    Student(){
        System.out.println("I am student constructor");
    }

    void setStudyClass(String sc){
        studyClass = sc;
    }

    void setName(String n){
        name = n;
    }

    void setRollno(int r){
        rollno = r;
    }
    void setPercentage(double p){
        percentage = p;
    }

    String getName(){
        return name;
    }

    String getStudyClass(){
        return studyClass;
    }

    int getRollno(){
        return rollno;
    }
    double getPercentage(){
        return percentage;
    }

}
```

Let’s represent a student

```
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        new Student();
    }
}
```

```
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
    }
}
```


Method Area
Heap Area
JVM Stack
PC
Native method stack
Per Thread
JVM Memory(Runtime data areas)
Class loader
Bytecode verifier
JIT Compiler
Interpreter
Execution Engine
Native method interface
Byte code
Garbage Collector
Security Manager
JVM
Native method libraries

```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
    }
}
```

21332730

Returns the memory location of that object

```java
String name;
String studyClass;
int rollno;
double percentage;
```

21332738

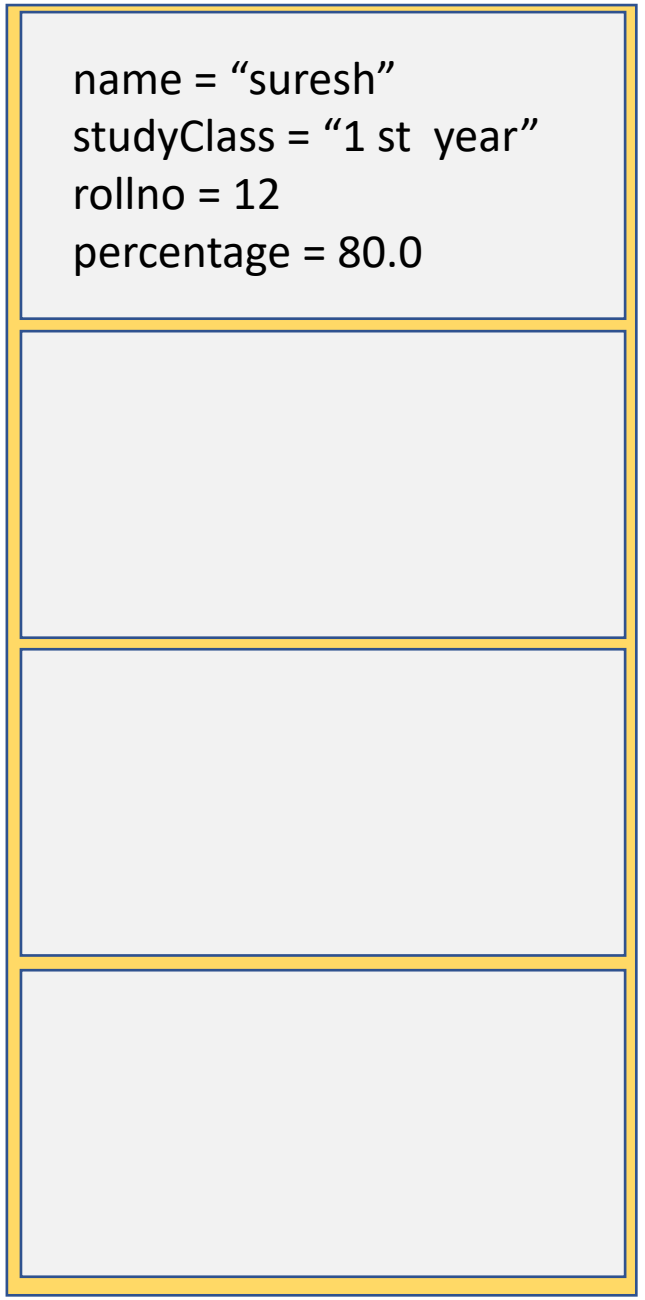
```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
        System.out.println(s.name);
        System.out.println(s.studyClass);
        System.out.println(s.rollno);
        System.out.println(s.percentage);
    }
}
```

21332746

```
Welcome to suresh techs college
Student@2c7b84de
null
null
0
0.0
```

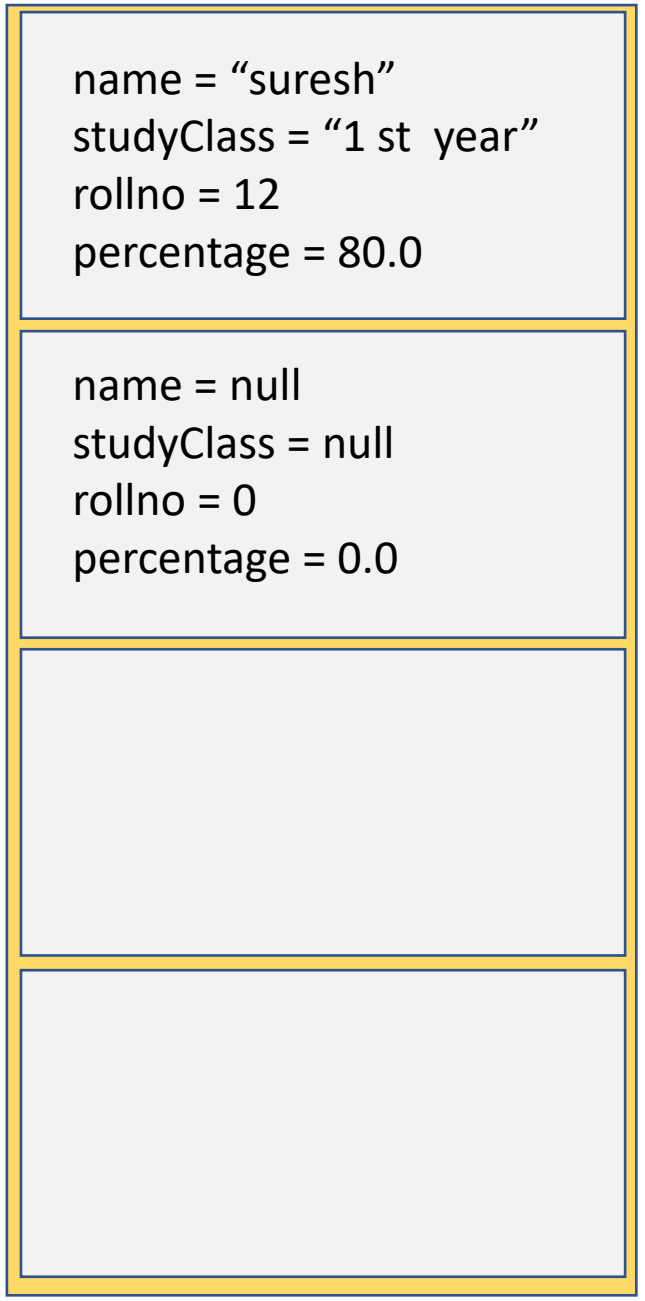
Memory

```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
        s.name = "suresh";
        s.studyClass = "1st year";
        s.rollno = 12;
        s.percentage = 80;
        System.out.println(s.name);
        System.out.println(s.studyClass);
        System.out.println(s.rollno);
        System.out.println(s.percentage);
    }
}
```



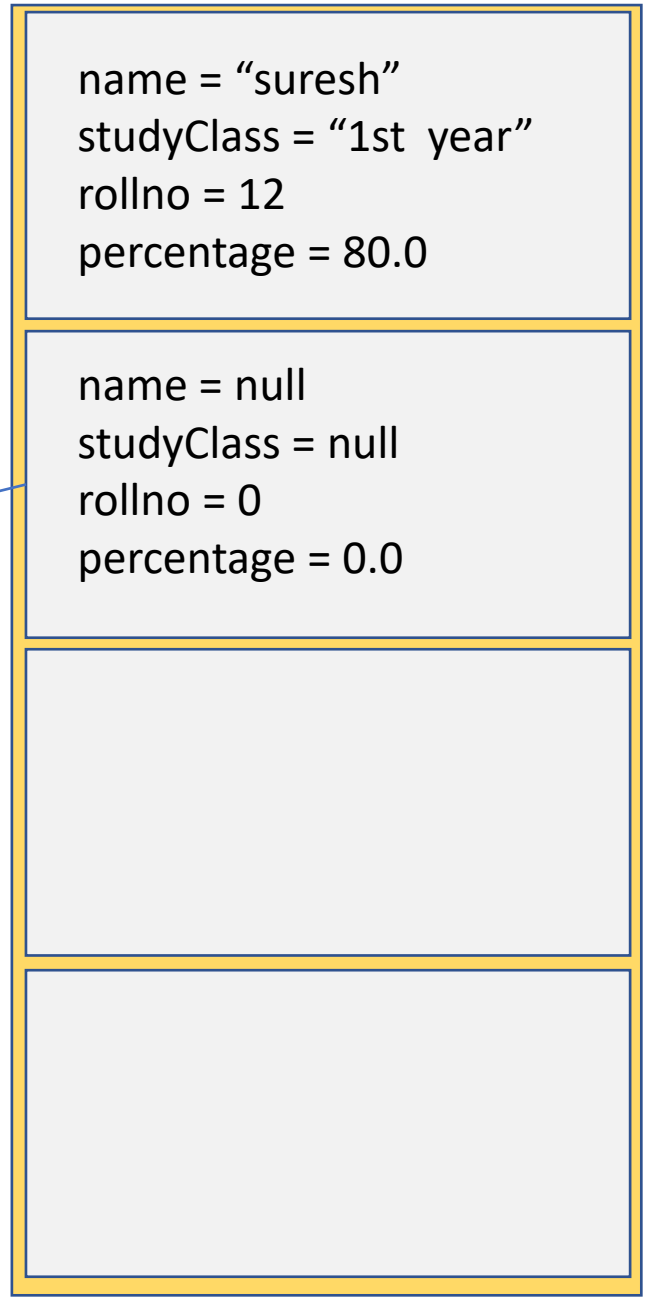
Memory

```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
        s.name = "suresh";
        s.studyClass = "1st year";
        s.rollno = 12;
        s.percentage = 80;
        System.out.println(s.name);
        System.out.println(s.studyClass);
        System.out.println(s.rollno);
        System.out.println(s.percentage);
        new Student();
    }
}
```



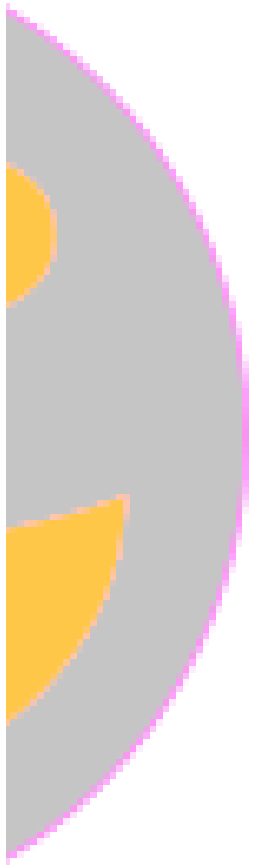
Memory

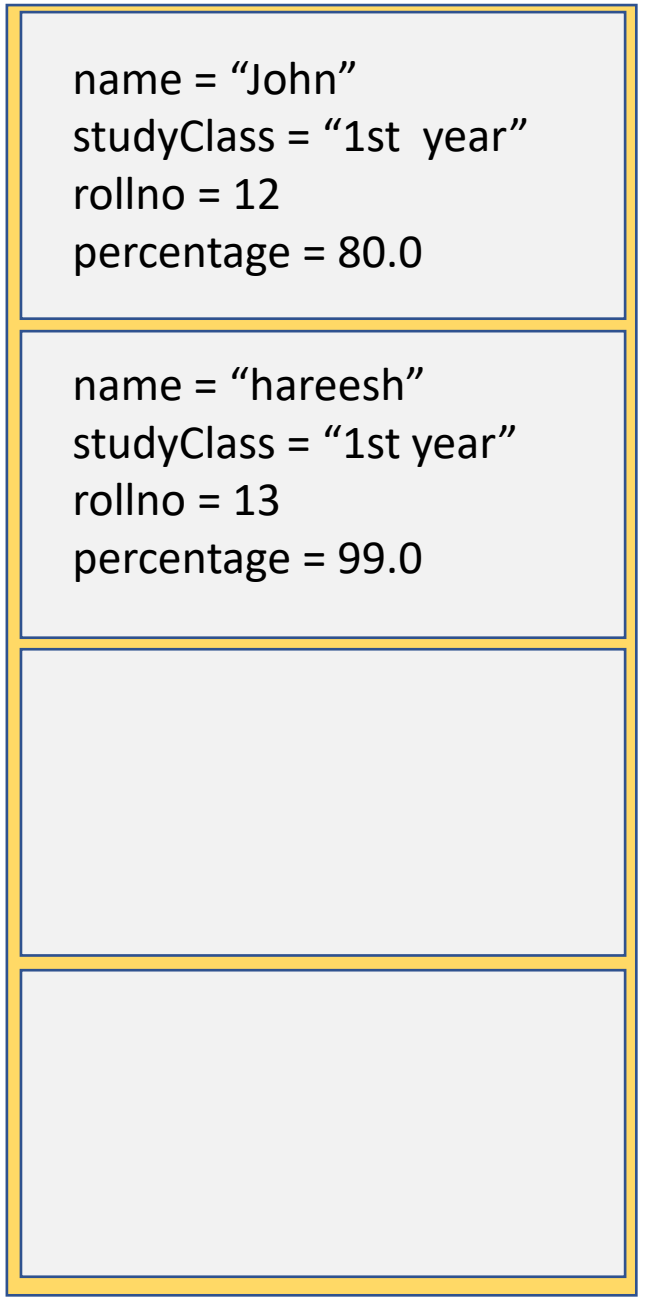
```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
        s.name = "suresh";
        s.studyClass = "1st year";
        s.rollno = 12;
        s.percentage = 80;
        System.out.println(s.name);
        System.out.println(s.studyClass);
        System.out.println(s.rollno);
        System.out.println(s.percentage);
        Student s1 = new Student();
    }
}
```



Memory

```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
        s.name = "suresh";
        s.studyClass = "1st year";
        s.rollno = 12;
        s.percentage = 80;
        System.out.println(s.name);
        System.out.println(s.studyClass);
        System.out.println(s.rollno);
        System.out.println(s.percentage);
        Student s1 = new Student();
        s1.name = "hareesh";
        s1.studyClass = "1st year";
        s1.rollno = 13;
        s1.percentage = 99;
        System.out.println(s1.name);
        System.out.println(s1.studyClass);
        System.out.println(s1.rollno);
        System.out.println(s1.percentage);
    }
}
```





Memory

Let’s use setters and getters

```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
        s.setName("suresh");
        s.setStudyClass("1st year");
        s.setRollno(12);
        s.setPercentage(80);
        System.out.println(s.name);
        System.out.println(s.studyClass);
        System.out.println(s.rollno);
        System.out.println(s.percentage);
        Student s1 = new Student();
        s1.name = "hareesh";
        s1.studyClass = "1st year";
        s1.rollno = 13;
        s1.percentage = 99;
        System.out.println(s1.name);
        System.out.println(s1.studyClass);
        System.out.println(s1.rollno);
        System.out.println(s1.percentage);
    }
}
```

```java
class Dashboard{
    public static void main(String[] args){
        System.out.println("Welcome to suresh techs college");
        Student s = new Student();
        System.out.println(s);
        s.setName("suresh");
        s.setStudyClass("1st year");
        s.setRollno(12);
        s.setPercentage(80);
        System.out.println(s.name);
        System.out.println(s.studyClass);
        System.out.println(s.rollno);
        System.out.println(s.percentage);
        Student s1 = new Student();
        s1.name = "hareesh";
        s1.studyClass = "1st year";
        s1.rollno = 13;
        s1.percentage = 99;
        System.out.println(s1.getName());
        System.out.println(s1.getStudyClass());
        System.out.println(s1.getRollno());
        System.out.println(s1.getPercentage());
    }
}
```

What is the use of constructor

- Used **to construct default state of an object**

```
Student(){
    percentage = 60;
    System.out.println("I am student constructor");
}
```

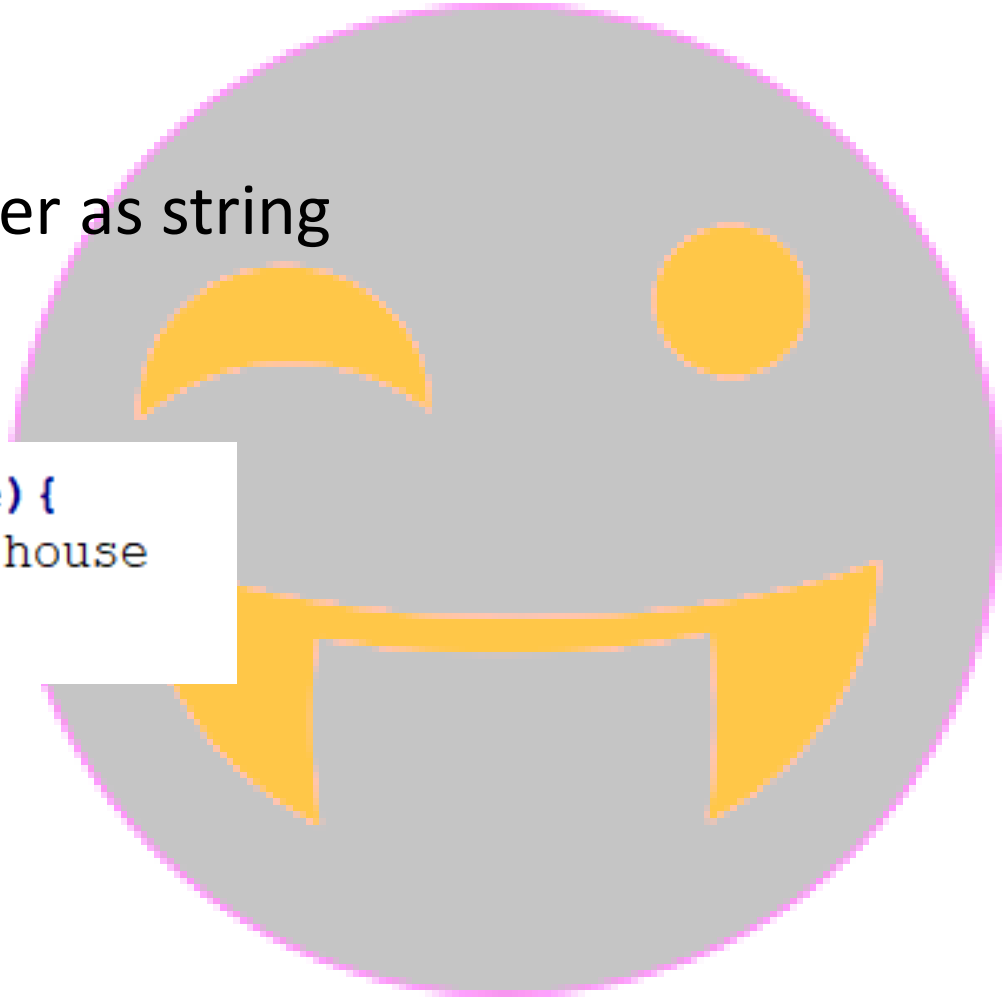
- We can assign some values to the state before creating an object

Parameterized constructors

- Constructor overloading

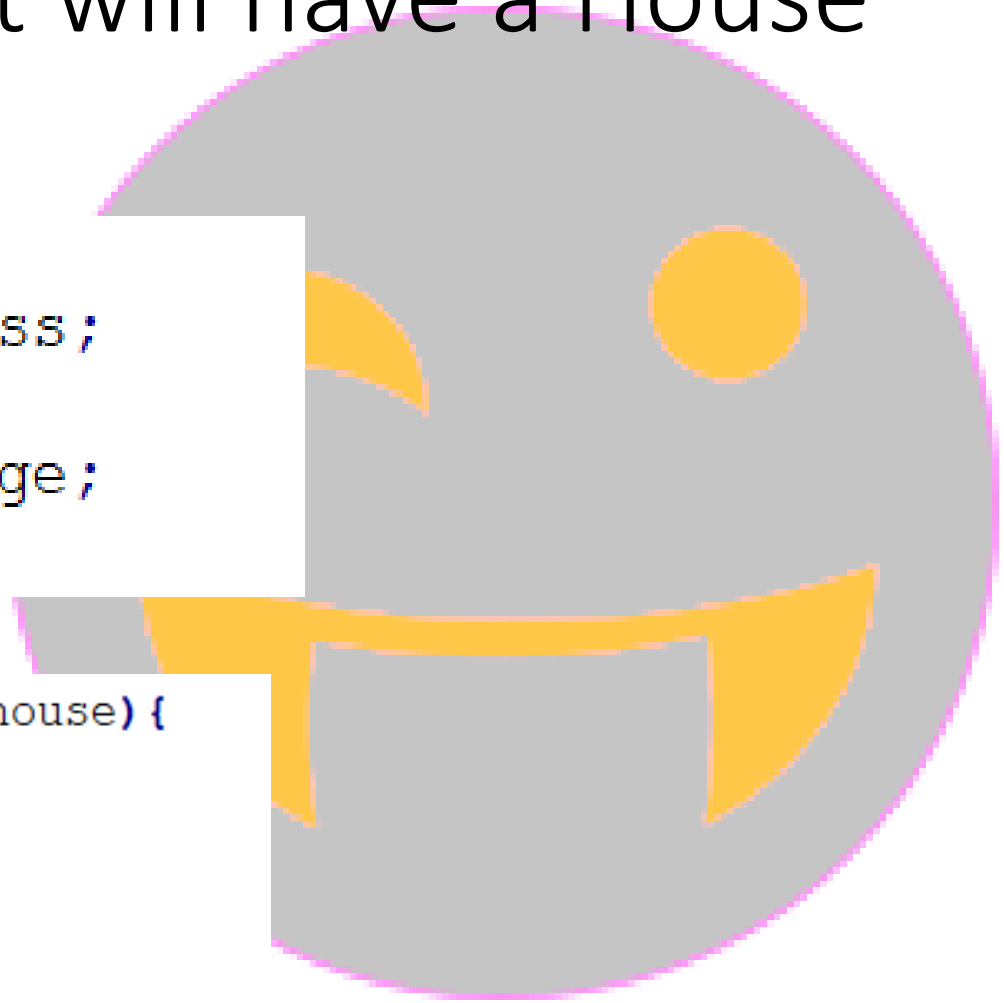
Create a parameterized constructor for House class

- Take house number as string



```
House(String house){
    houseNumber = house
}
```

Each student will have a House

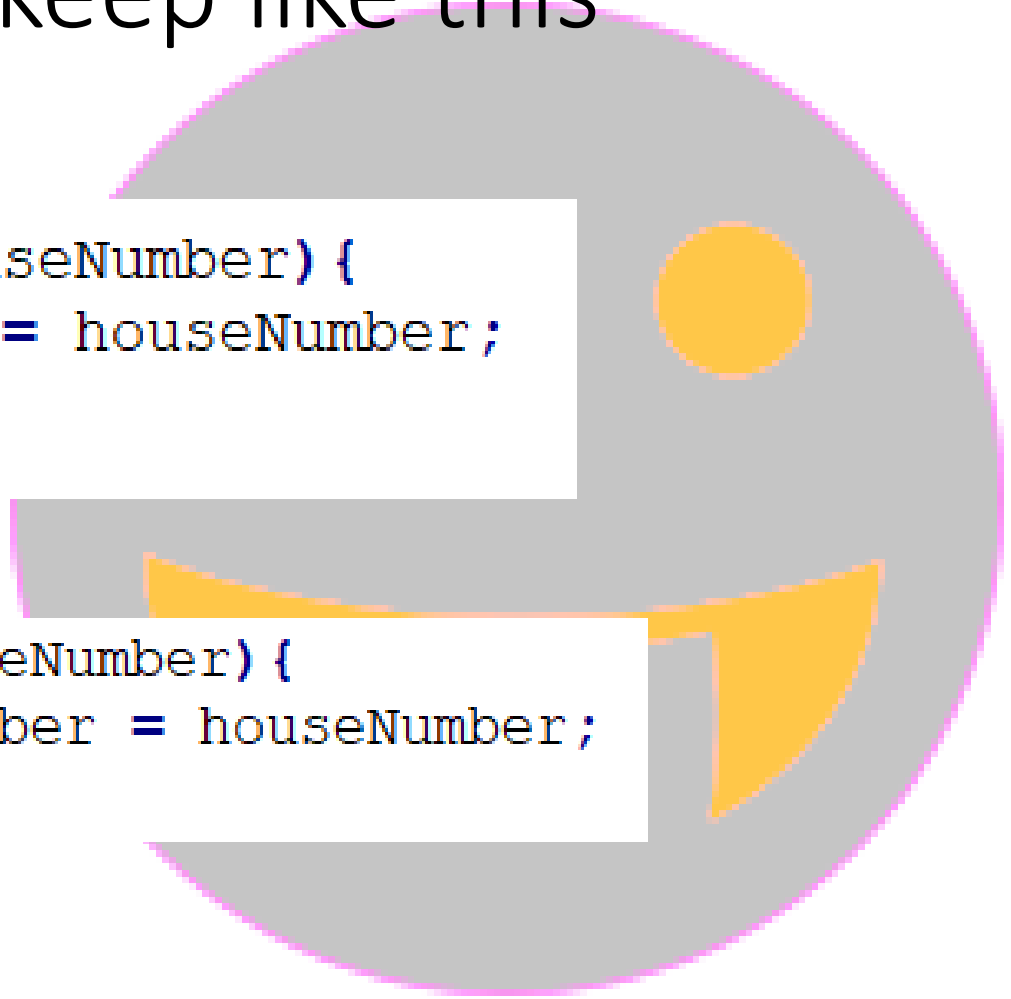


```
String name;
String studyClass;
int rollno;
double percentage;
House h;


void setHouse(House house){
    h = house;
}

House getHouse(){
    return h;
}
```

What if you keep like this

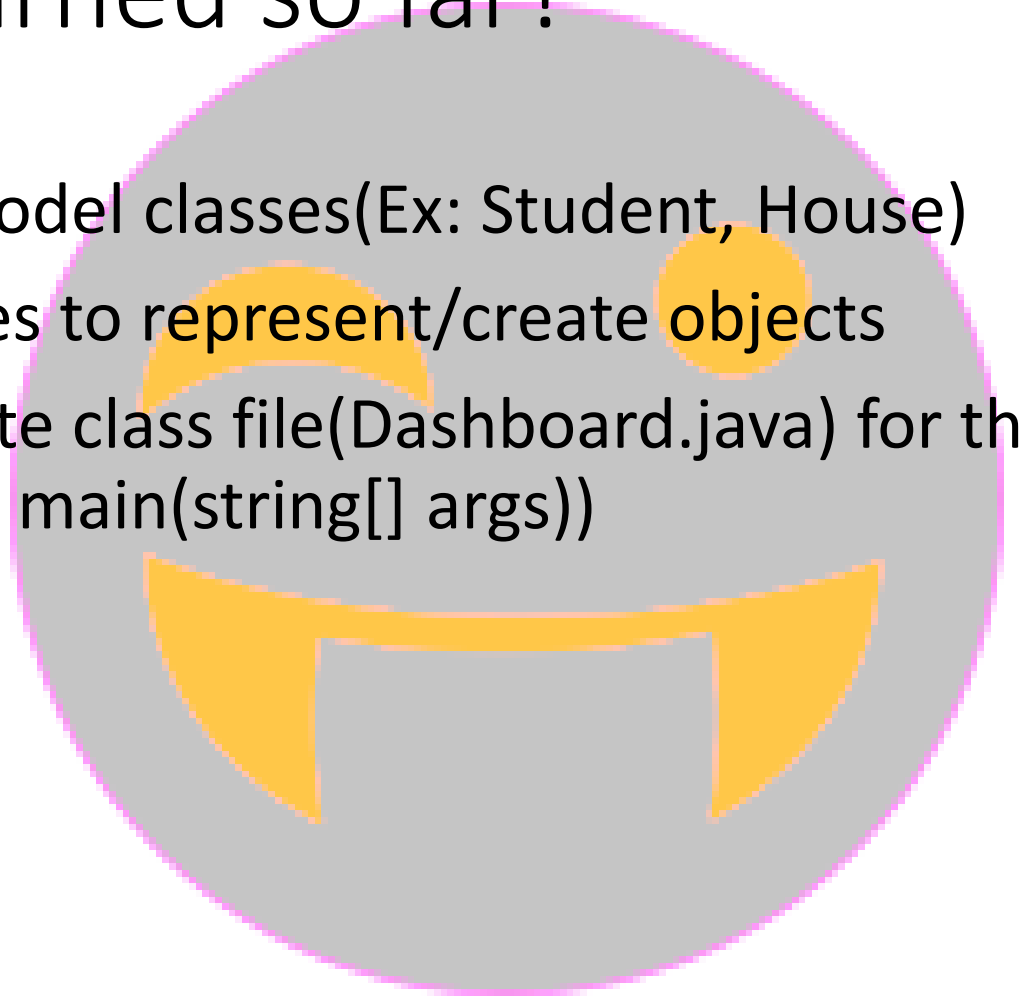
```
House(String houseNumber){
    houseNumber = houseNumber;
}
```

```
House(String houseNumber){
    this.houseNumber = houseNumber;
}
```



What we learned so far?

- Creating POJO/Model classes(Ex: Student, House)
- Using POJO classes to represent/create objects
- Creating a separate class file(Dashboard.java) for the entry point (public static void main(string[] args))



What is Java?

- Desktop applications
- Mobile applications
- Web applications
- Enterprise applications
- Embedded systems
- Bigdata technologies
- Etc..

2. High level language
4. General-Purpose
3. Object Oriented (99%)

: is a high level, general-purpose, class
object-oriented programming langua

Invented a complete new language

Java

*7

Oracle acquired sun microsystems in 2010, January 27 for US $7.4 billion

ogram ?
ogram is a set of instructions
sed to perform a task
order to develop an

Suresh Techs Software Solution

Application, is short app, is a software that perform specific tasks for an end us

Software Engineers
Employees

ject to develop
cation

Difference between
program, package,
library & application

Source code
javac
Addition.java

Enter first number
190
Enter second number
1999

resh techs, I am le
resh, I will get jo

s\" is 5 star");

1. Class loa
2. Runtim
3. Execut

javac filename.java
Compiler
Addition.class
java classname
Interpreter
Results

Byte code
Execution Engine
Garbage Collector
Security Manager
JVM
Native method libraries

Loads the class file and executes it

17 TOTAL COURSE
18 TOTAL BATCH
JO classes
1 Total Admin Users
To do's
Subject

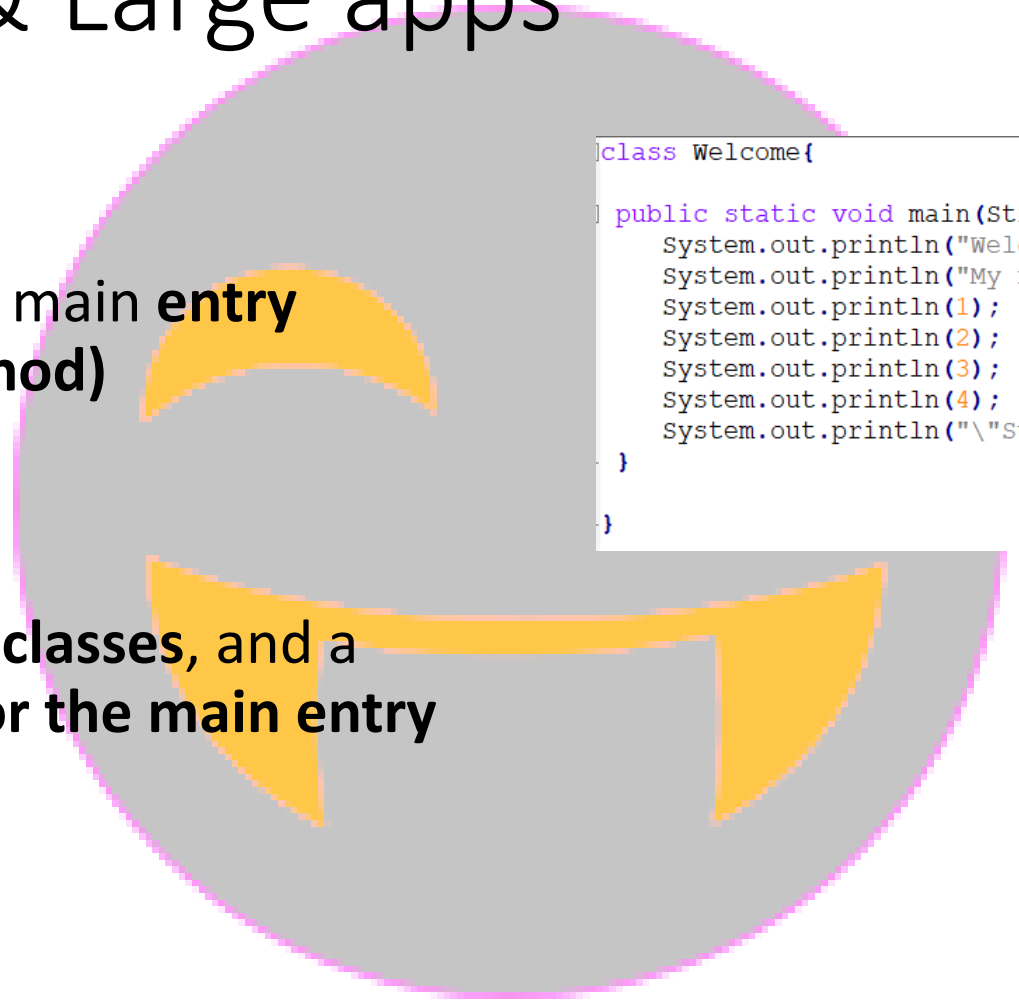
Data types & Met

Single line
Multi line
Documentation comments

Using Model classes, Creating Objects, Constructors, Methods, Methods overloading, local/instance variables etc.

Small apps & Large apps

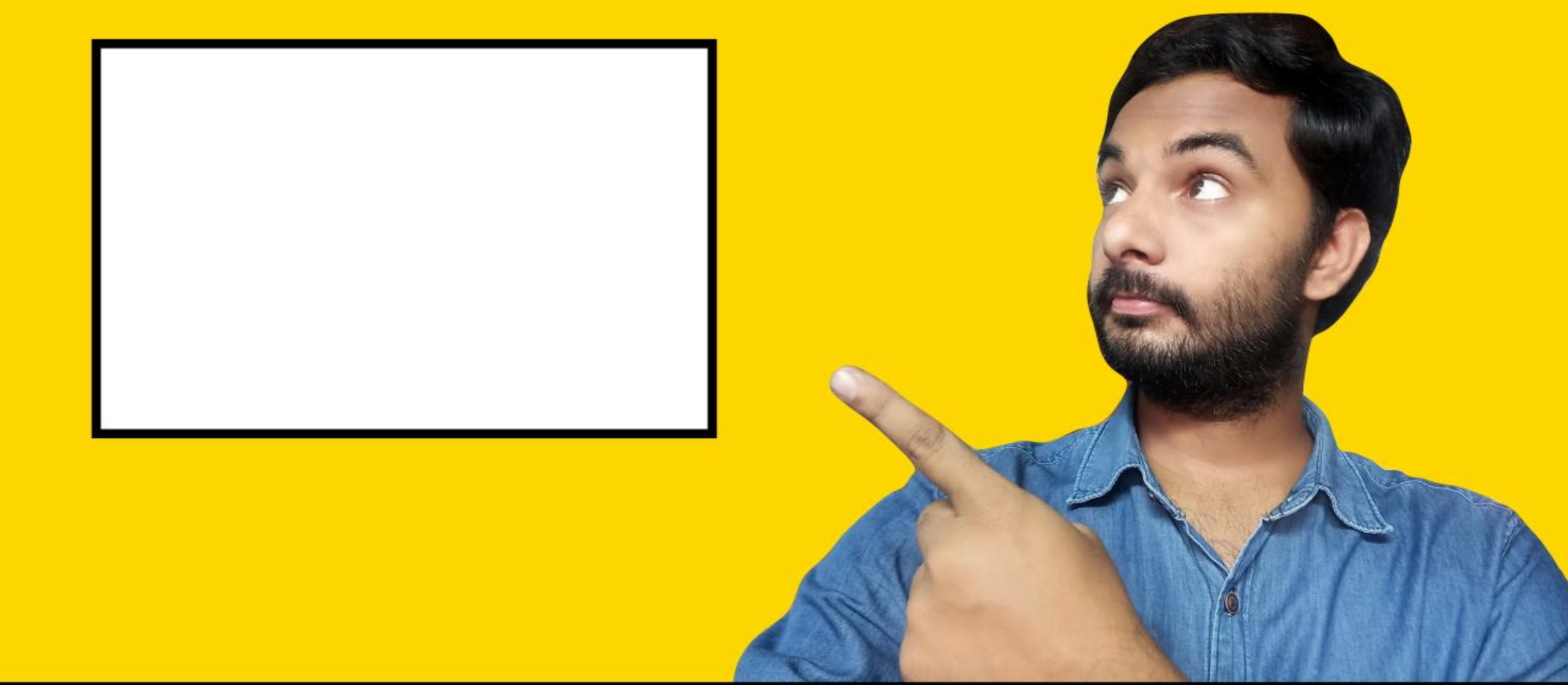
- Small apps
  - **Single class** with main **entry point(main method)**
- Large apps
  - Different **Model classes**, and a **separate class for the main entry** point



```
class Welcome{

 public static void main(String[] args){
    System.out.println("Welcome to suresh techs, I am learning Java.");
    System.out.println("My name is suresh, I will get job soon");
    System.out.println(1);
    System.out.println(2);
    System.out.println(3);
    System.out.println(4);
    System.out.println("\"Suresh techs\" is 5 star");
 }

}
```

What next?

Variables in detail



చిన్న బ్రేక్ చిటికలో వచ్చేస్తా