CLASSES : INTRODUCTION

A CLASS IS A TYPE OF VARIABLE

ANY PROGRAMMER CAN COME UP WITH CLASSES

I.E. ANY PROGRAMMER CAN DEFINE AS MANY NEW TYPES OF VARIABLES AS SHE LIKES

THEN, OTHER PROGRAMMERS CAN CREATE VARIABLES OF THAT CLASS

VARIABLES OF A CLASS ARE CALLED "OBJECTS" OF THAT CLASS

(JUST AS THEY WOULD CREATE VARIABLES LIKE LISTS OR DICTIONARIES)

"OBJECTS" AND "CLASSES" ARE SUPER-SUPER IMPORTANT CONCEPTS IN COMPUTER SCIENCE

Class definition - this is how you create your own class, its pretty bare bones and useless at this point

```
public class Person {

}
```

STRINGS, LISTS AND DICTIONARIES ARE CLASSES TOO

NUMBERS, STRINGS, LISTS AND DICTIONARIES ARE CALLED **"BUILT-IN" TYPES**

(BECAUSE VARIABLES OF THESE TYPES CAN BE CREATED RIGHT OUT OF THE BOX)

STRINGS, LISTS AND DICTIONARIES ARE OBJECTS TOO, I.E. THEY ARE VARIABLES CREATED FROM TYPES THAT ARE CLASSES

Classes, variables and objects

```java
// Person is a class "mark" and "tom" are variables which hold objects of the class Person
Person mark = new Person();
Person tom = new Person();

// String is a class, "s" is a variable which holds an object of class String
String s = "Mark Twain";

// Boolean is a class "b" is a variable which holds an object of class Boolean
Boolean b = false;

// Integer is a class, "i" is a variable which holds an object of class Integer
Integer i = 5;
```

INSTANTIATION

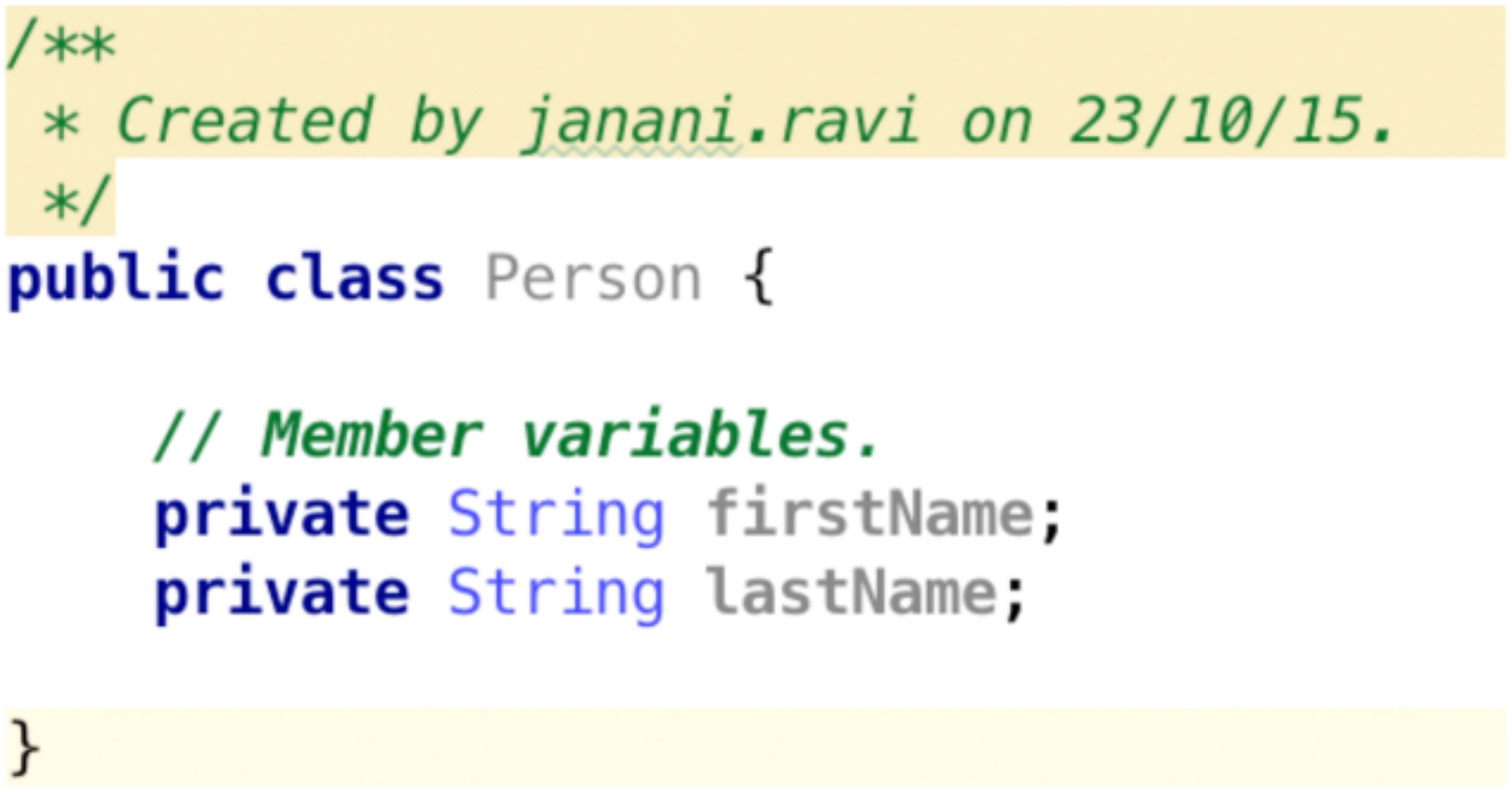
CREATING A VARIABLE OF A TYPE
IS CALLED INSTANTIATION

SIMILARLY

CREATING AN OBJECT OF A CLASS
IS CALLED INSTANTIATION

"OBJECTS ARE INSTANCES OF THEIR CLASS"

Person class with member variables - the member variables can be of any type



```java
/**
 * Created by janani.ravi on 23/10/15.
 */
public class Person {

    // Member variables.
    private String firstName;
    private String lastName;

}
```

ENCAPSULATION

OBJECTS CONTAIN WITHIN THEM BOTH

DATA AND FUNCTIONS

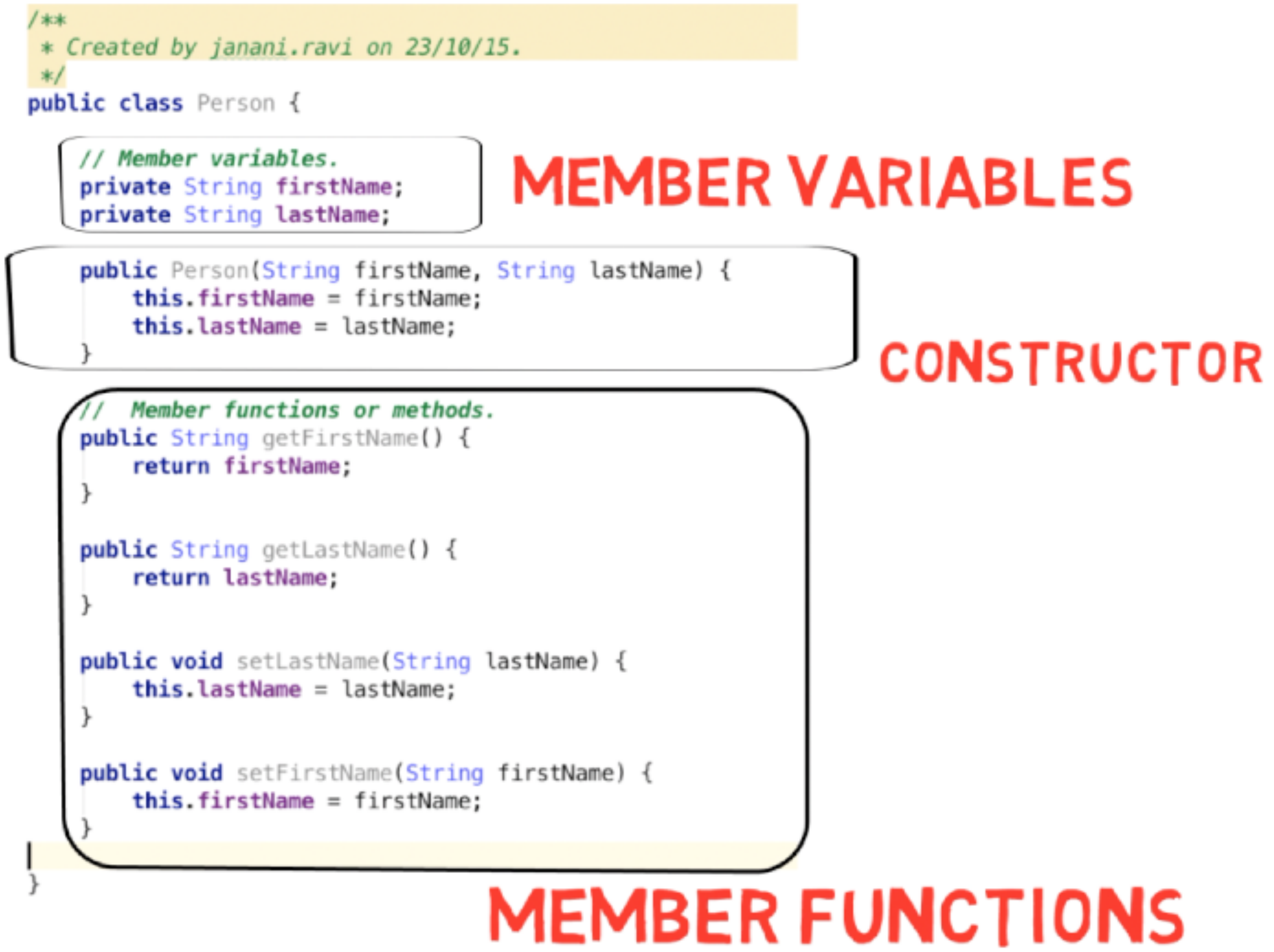
DATA VARIABLES CONTAINED INSIDE AN OBJECT ARE CALLED

MEMBER VARIABLES

FUNCTIONS CONTAINED INSIDE AN OBJECT ARE CALLED

METHODS, OR MEMBER FUNCTIONS

THUS OBJECTS ARE SELF-CONTAINED - THEY CARRY AROUND BOTH THE FUNCTIONS AND THE DATA NEEDED TO DO STUFF WITH THEM



```java
/**
 * Created by janani.ravi on 23/10/15.
 */
public class Person {

    // Member variables.
    private String firstName;
    private String lastName;

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    //  Member functions or methods.
    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

}
```

WITH GREAT POWER

THE ABILITY TO CREATE CLASSES
IS GREAT POWER

THAT COMES WITH THE RESPONSIBILITY
TO SET UP MEMBER VARIABLES CORRECTLY,
WHEN THE OBJECT IS BORN, AND CLEAN
THEM UP WHEN THE OBJECT GOES AWAY

COMES GREAT RESPONSIBILITY

SETUP AND CLEANUP

OBJECTS CONTAIN DATA AND FUNCTIONS

THOSE DATA - MEMBER VARIABLES - CAN BE STRINGS, LISTS, DICTIONARIES, FILES, DATABASE CONNECTIONS, OR OTHER OBJECTS

DATABASE CONNECTIONS NEED TO BE OPENED, CLOSED AND POSSIBLY COMMITTED

FILES NEED TO BE OPENED AND CLOSED

CONSTRUCTORS

ARE SPECIAL METHODS (MEMBER FUNCTIONS) THAT ARE CALLED AUTOMATICALLY WHEN AN OBJECT IS CREATED

CONSTRUCTORS ALWAYS HAVE THE SAME NAME AS THE CLASS THAT THEY BELONG IN

FINALIZERS

ARE SPECIAL MEMBER FUNCTIONS CALLED WHEN AN OBJECT IS ABOUT TO CEASE TO EXIST