



Computer Science Virtual Learning

HS Computer Science A

April 9th, 2020

Lesson: **Parts of a Java Class**

Objective/Learning Target:

1. Students will be able to identify the differences between methods, fields, constructors and variables within a Java class.



Activity

A Java class defines what objects of the class know (fields) and can do (methods). The class also defines how to initialize the fields when the object is first created (constructors).

```
//////////////// main for testing //////////////////  
public static void main(String[] args)  
{  
    Person p1 = new Person("Deja", "555 132-3253");  
        System.out.println(p1);  
        Person p2 = new Person("Avery", "555  
132-6632");  
        System.out.println(p2);  
}
```



Activity - Part 1: Methods and Running Your Java Program

When you click the **Run** button Java starts execution in the main method as shown in the code below (public static void main(String[] args)). The body of the main method is all the code between the first { and the last }. Every class in Java can have a main method.

If you are executing a Java program using the command line tools you will type java
ClassName and execution will start in the specified class's main method.

1. In the code below, highlight/circle the code that would be considered the body.

```
////////////////// main for testing ////////////////////  
  
public static void main(String[] args)  
{  
  
    Person p1 = new Person("Deja", "555 132-3253");  
  
    System.out.println(p1);  
  
    Person p2 = new Person("Avery", "555 132-6632");  
    System.out.println(p2);  
  
}
```

*****Note: A main method should create the objects that will do the work in an object-oriented program then ask the objects to do the work**



Activity - Part 1: Methods and Running Your Java Program

The previous code is the main method for the Person class. It shows two variables (p1 and p2) of type Person being created and each of the variables refers to a new Person object. Each new Person object's name and phone number are set to the passed values (new Person("Deja", "555 132-3253")). Then each object's toString method is called to output information about the object. The toString method is called when you try to print the value of an object using `System.out.println(object);`.

It's common to have a main method in each of my classes that tests the methods in that class. The main method in the Person class creates two Person objects and prints their values out using the toString method. The toString method is what is called on an object when you execute `System.out.println(object)`.



Activity - Part 1: Methods and Running Your Java Program

The main method must be declared as `public static void main(String[] args)`. The only part of this that you can change is the `args`. You can use a different name instead if you wish. The `public` keyword is necessary since this method needs to be executed from outside the current class. The `static` keyword means that you can execute this method on the class (not on an object), which is important since no objects of this class have been created yet when the main method starts. The `void` keyword says that this method doesn't return anything. The `(String[] args)` says that this method can take some information when you execute it which will be passed to the method as an array of strings called `args`.



Check Your Understanding - Part 1

Click the link below to take you to runestone academy and scroll down to the bottom of the page

<https://runestone.academy/runestone/books/published/apcsareview/JavaBasics/runClass.html>

Try changing the code in the main method (`public static void main(String[] args)`) that creates as new Person object `new Person("Deja", "555 132-3253")` to create an object to represent you and one of your friends (but don't use your actual phone numbers). Click the **Run** button to test your code. An example of an output is shown

```
name: Deja, cell: 555 132-3253  
name: Avery, cell: 555 132-6632
```



Activity - Part 2: Instance Variables

Fields hold the data for an object. Fields record what an object needs to know to do work in the program. Fields are also called instance variables or object variables or properties.

All fields on the AP CS A exam should be declared private. Think of private as like your diary. Only you should have direct access to it. In this case private means that only the code in this class can directly access the field values.

*****Note:** Fields are declared right after the class declaration. They start with private then the *type* of the field and then a *name* for the field.



Activity - Part 2: Instance Variables

The Person class declares two fields: name and cell. Name is the person's name and cell is their cell phone number. These are both things that you might need to know about a person.

```
/// fields //////////////////////////////////  
private String name;  
private String cell;
```



Check Your Understanding - Part 2

Highlight/circle all of the field declarations in the following class.

```
public class Name {  
    private String first;  
    private String last;  
    public Name(String theFirst, String  
theLast) {  
        first = theFirst;  
        last = theLast;  
    }  
    public void setFirst(String theFirst) {  
        first = theFirst;  
    }  
    public void setLast(String theLast) {  
        first = theLast;  
    }  
}
```



Activity - Part 3: Methods

Methods define what an object can do or the behavior of the object.

Most methods you work with will be public.

****Note: Methods define what the object can do. Methods are specified after the constructors. They typically start with public then a type, then the name of the method. They can take data as input which is specified in parentheses.**

The Person class has methods for getting the name and cell phone and for setting the cell phone.

Methods that get information from an object are called **getters** or **accessors**. Methods that set field values are called **setters** or **mutators**.



Check Your Understanding - Part 3

1. Highlight/circle the getter methods in the code above
2. Highlight in a different color/square the setter methods in the code above

```
public String getName()  
{  
    return this.name;  
}  
public void setName(String theName)  
{  
    this.name = theName;  
}  
public String getCell()  
{  
    return this.cell;  
}  
public void setCell(String theCell)  
{  
    this.cell = theCell;  
}  
public String toString() { return "name:  
" + this.name + ",  
cell: " + this.cell; }
```



For More Resources and to Check Answers

Go to: <https://runestone.academy/runestone/books/published/apcsareview/JavaBasics/partsOfAClass.html>