Classes, Objects, and Constructors

This guide has four subsections, namely:

2.6.1 Creating a class

2.6.2 Creating an object

2.6.3 Creating a constructor

2.6.4 Pushing the code to your GitHub repositories

**Step :** Creating a class

* A class is a blueprint from which individual objects are created.

**public** **class** Dog {

**String** breed;

int age;

**String** color;

void barking() {

}

void hungry() {

}

void sleeping() {

}}

**Step :** Creating an object

* If we consider the real world, we can find many objects around us, such as cars, dogs, humans, and so on. All these objects have a state and behavior.
* If we consider a dog, then its state is: name, breed, color, and the behavior is: barking, wagging the tail, and running.
* If you compare the software object with a real-world object, they have very similar characteristics.
* Software objects also have a state and behavior. A software object's state is stored in fields and behavior is shown via methods.
* So, in software development, methods operate on the internal state of an object and the object-to-object communication is done via methods.

**public** **class** Puppy {

**public** Puppy(**String** name) {

// This constructor has one parameter, name.

**System**.out.println("Passed Name is :" + name );

}

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

// Following statement would create an object myPuppy

Puppy myPuppy = **new** Puppy( "tommy" );

}}

**Step :** Creating a constructor

* When discussing about classes, one of the most important subtopics would be constructors. Every class has a constructor. If we do not explicitly write a constructor for a class, the Java compiler builds a default constructor for that class.
* Each time a new object is created, at least one constructor will be invoked. The main rule of constructors is that they should have the same name as the class. A class can have more than one constructor.

**public** **class** Puppy {

**public** Puppy() {

}

**public** Puppy(**String** name) {

// This constructor has one parameter, name.

}}

**Step :** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master