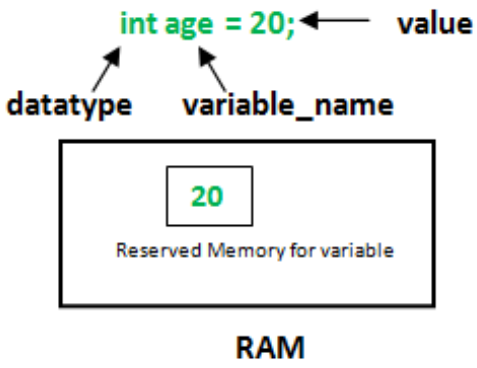
Variables in Java

We can say variable is a container which can store the value within it.

e.g., int age = 20;



There are mainly three types of variables in java:

1. Local variables
2. Class variables
3. Instance variables

Local Variables

- Variables which are declared inside a method, constructor or block are known as local variables.
- Local variables are created when control of a program enters in a method, constructor or block and destroys when program control exits the method.
- We cannot use any access specifier with local variables.
- Local variables are available to use only within the method, constructor or block where they are declared.
- There is no default value for local variables.
- Local variables must be initialized before use.

```
public class LocalVariable {

   public void printLocalVariable() {
         int localVariable = 10;
         System.out.println(localVariable);
   }
}
```

As you can see in the above code, the **localVariable** is declared inside a method. That is why it is considered as local variable.

Class Variable

- Variables which are declared within a class but not inside a method, constructor or block.
- There can be only one copy of a class variable throughout the class regardless of how many instances we create from that class.
- These variables are created when program execution starts and destroys when program execution stops.
- Class variables are available for every method, constructor or block present in the class where they are declared.
- Class variables have default value and it varies based on the data type. For int, short, long, byte it is 0. For float 0.0f, for double 0.0d, for Boolean it is false, for char it is '\u0000'.
- We can use access specifiers with class variables.

```
public class ClassVariable {

   public int classVariable = 10;

   public void printClassVariable() {

        System.out.println(classVariable);
   }

   public static void main(String[] args) {
        ClassVariable classVariable = new ClassVariable();
        classVariable.printClassVariable();
   }
}
```

Output of Program:

10

As you can see in the above code the **classVariable** is the variables declared inside the class but outside of the method. That is why it is considered as class variable.

Instance Variables

- Similar to class variables, instance variables are declared inside a class but outside of a method, constructor or block.
- When space is allocated for an object at the same time instance variables are created inside it.
- Instance variables are created when objects are created with the use of new keyword and destroyed when the object is destroyed.
- Instance variables contains values which is available for all methods, constructors or blocks.
- We can specify access specifiers to instance variables.
- Instance variables have default values. For numbers, the default value is 0, for boolean it is false, and for object references it is null. Values can be assigned during the declaration or within the constructor.
- Instance variables are associated to object. They can' t be marked as static.
- Their values are unique to each instance of a class.

```
public class Caffe {
//These values are unique to each instance we made of this class
   public String name;
   public double price;

   public Caffe() {
         this.name = "Coffee";
         this.price = 80.00;
   }

   public void printProductAndPrice() {
         System.out.println("Product :"+name+" "+"Price :"+price);
   }
   public static void main(String[] args) {
         Caffe caffe = new Caffe();
         caffe.printProductAndPrice();
   }
}
```

As you can see in the above code the variables name and price are declared within a class Caffe. This name and price will be different for different objects we create from class Caffe.

```
    public Caffe() {
            this.name = "Coffee";
            this.price = 80.00;
        }
```

Above code is constructor. We will learn this in coming lectures.