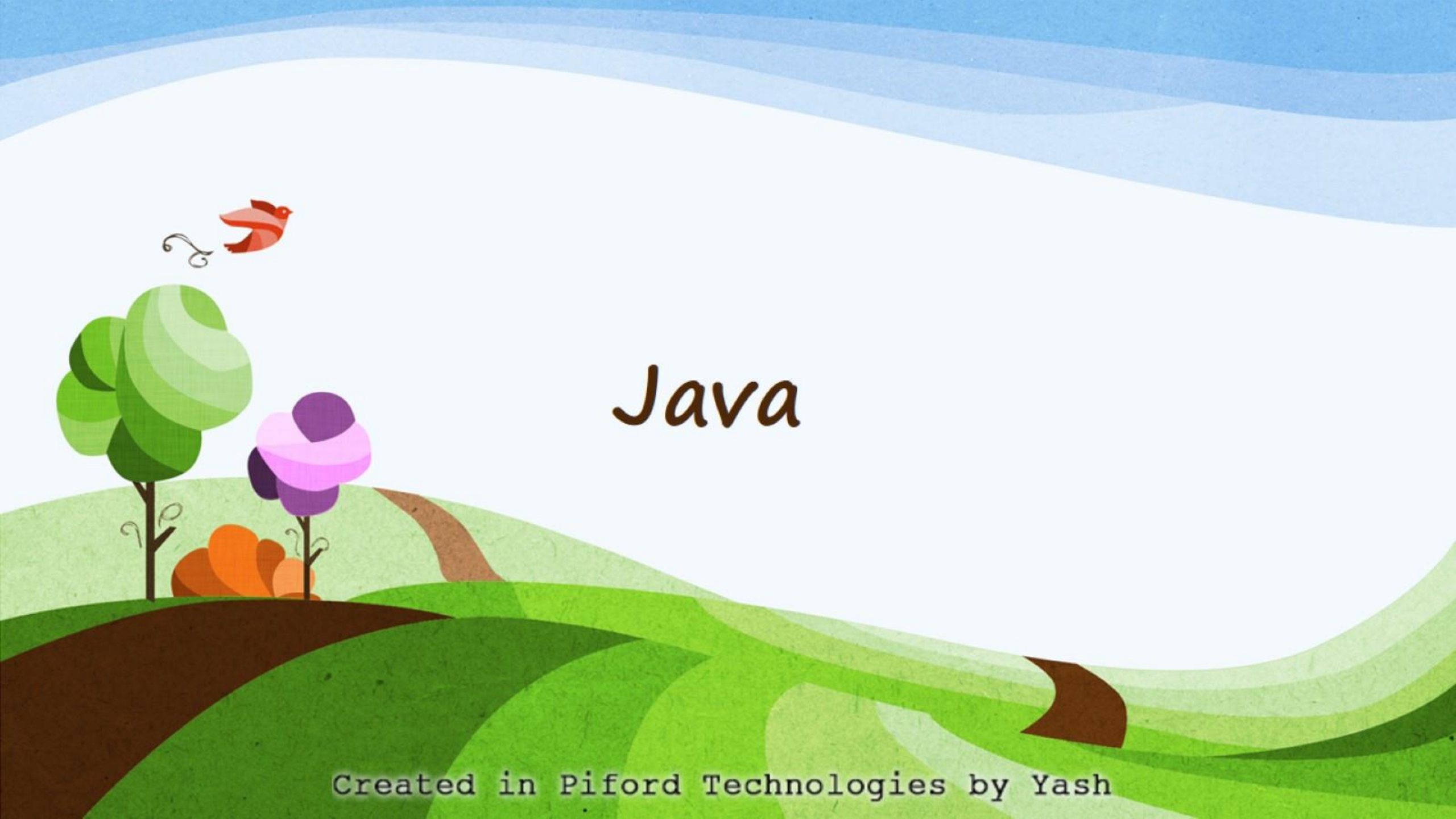
Java
Created in Piford Technologies by Yash

Constructor

Created in Piford Technologies by Yash

Constructor

- Constructor in java is a special type of method that is used to initialize the object.
- Java constructor is invoked at the time of object creation.
- It constructs the values i.e. provides data for the object that is why it is known as constructor.

Rules for creating java constructor

- There are basically two rules defined for the constructor.
  - Constructor name must be same as its class name
  - Constructor must have no explicit return type

Types of java constructors

- There are two types of constructors:
  - Default constructor (no-arg constructor)
  - Parameterized constructor

Default Constructor

- A constructor that have no parameter is known as default constructor.

Example:

Hornet.java

```
1  package try5;
2
3
4
5  class Hornet {
6          Hornet(){
7              System.out.println("We are inside a constructor");
8          }
9      public static void main(String[] args) {
10             Hornet h = new Hornet();
11     }
12 }
13
```

Parameterized constructor

- A constructor that have parameters is known as parameterized constructor.

Example:

*Hornet.java

```
1  package try5;
2
3  class Hornet {
4      String color;
5        Hornet(String c){
6            color=c;
7        }
8        void show(){
9            System.out.println(color);
10       }
11     public static void main(String[] args) {
12           Hornet h = new Hornet("Orange");
13           h.show();
14     }
15 }
16   Created in Piford Technologies by Yash
```

Constructor Overloading

- Constructor overloading is a technique in Java in which a class can have any number of constructors that differ in parameter lists.
- The compiler differentiates these constructors by taking into account the number of parameters in the list and their type.

Example:

Hornet.java

```
1  package try5;
2
3  class Hornet {
4      String color;
5      int cc;
6        Hornet(String c){
7            color=c;
8        }
9        Hornet(String c, int ccc){
10           color=c;
11           cc=ccc;
12       }
13       void show(){
14           System.out.println(color+" "+cc);
15       }
16     public static void main(String[] args) {
17           Hornet h = new Hornet("Orange");
18           Hornet h2 = new Hornet("Orange",160);
19           h.show();
20           h2.show();
21     }
22 }
```

Output: Orange 0
Orange 160

Methods

Created in Piford Technologies by Yash

Methods

- A Java method is a collection of statements that are grouped together to perform an operation.
- For example when you call the System.out.println() method, the system actually executes several statements in order to display a message on the console.

Creating Method

Syntax:

```
modifier returnType nameOfMethod (Parameter List) {
   // method body
}
```

The syntax shown includes:

- **Modifier** – It defines the access type of the method and it is optional to use.
- **ReturnType** – Method may return a value.
- **NameOfMethod** – This is the method name. The method signature consists of the method name and the parameter list.
- **Parameter List** – The list of parameters, it is the type, order, and number of parameters of a method. These are optional, method may contain zero parameters.
- **Method body** – The method body defines what the method does with the statements.

Example:

```
public static int methodName(int a, int b) {
   // body
}
```

Method Calling

- For using a method, it should be called.
- There are two ways in which a method is called:
  - Method returns a value
  - Returning nothing

Example Method returns a value:

Hornet.java

```
1  package try5;
2
3  class Hornet {
4      public static void main(String[] args) {
5        int a=9;
6        int b=2;
7        int c=check(a,b);
8        System.out.println(c);
9      }
10     public static int check(int x,int y){
11         int z=x+y;
12         return z;
13     }
14 }
15
```

Example Method Returning nothing:

Hornet.java

```
1  package try5;
2
3  class Hornet {
4      public static void main(String[] args) {
5        int a=9;
6        int b=2;
7        check(a,b);
8
9      }
10     public static void check(int x,int y){
11         int z=x+y;
12         System.out.println(z);
13     }
14 }
15
```