Java has the concept of class and object. In order to access the code written inside a particular class, we need to create an object of that class and then new call the code. That’s why Java is OOP.

Why Java is not 100% object-oriented?

Ans: Java has its primitive data type. Along with class objects it supports primitive datatypes in order to declare variables. That’s why Java is not 100% OOP.

Class A

1. Method1
2. Method2

Class B -> child of class A

1. Method1
2. Method2

B obj

Obj.Method1

Obj.Method2

Basically, we can access the code present inside the parent class from the object of the child class.

Class Parent A Class Parent B

Method1 Method2

Class Child C -> A, B

Method1 and Method2

1. Method Overriding -> Run time polymorphism
2. Method Overloading -> Compile time polymorphism

Constructor -> A snippet of code that will be called when the object of the class is getting created. The name of the constructor is as same as the class name.

The method has a return type but the constructor does not.

The method needs to be called but the constructor gets called automatically when creating an object of the class.

The class name should be the same as the Constructor name, but methods don’t have the same name as the class.