A Bit More about Constructors

In this lesson, you will learn a bit more about constructors.

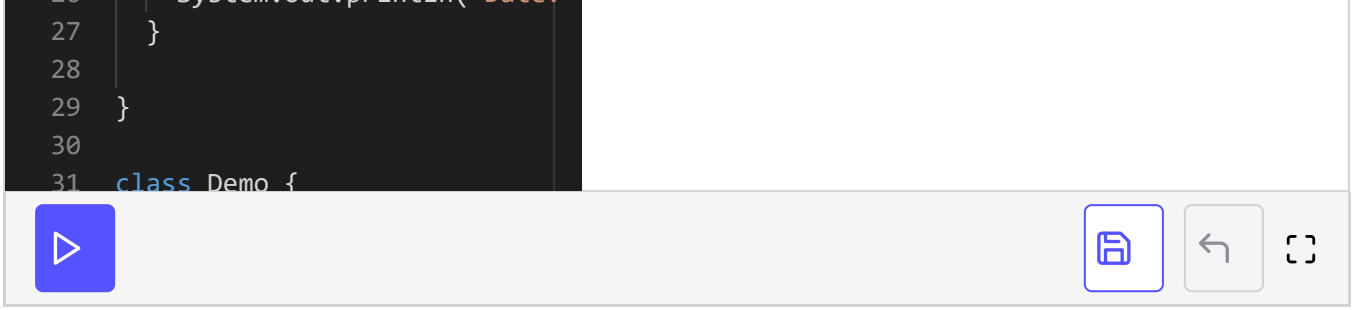
We'll cover the following

`this` Reference Variable

The `this` reference variable exists for every class. It refers to the class object itself. We use the `this` when we have an argument which has the same name as a data member. `this.memberName` specifies that we are accessing the `memberName` variable of the particular class.

Let's see it in action:

```
class Date {

  private int day;
  private int month;
  private int year;


  // Default constructor
  public Date() {
    // We must define the defau
    this.day = 0;
    this.month = 0;
    this.year = 0;
  }

  // Parameterized constructor
  public Date(int day, int mont
    // The arguments are used a
    this.day = day;
    this.month = month;
    this.year = year;
  }

  // A simple print function
  public void printDate(){
    System.out.println("Date: "
```



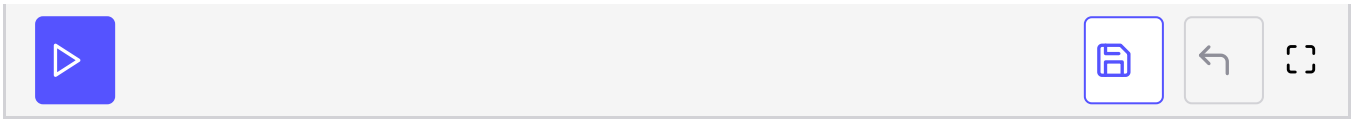
Calling a Constructor from a Constructor

In Java, we can call a constructor from a constructor. When you call a constructor from another constructor, you use the `this` keyword to refer to the constructor.

Let's see it in action:

```
class Date {

  private int day;
  private int month;
  private int year;
  private String event;


  // Default constructor
  public Date() {
    // We must define the default values for day, month, and year
    this.day = 0;
    this.month = 0;
    this.year = 0;
  }

  // Parameterized constructor
  public Date(int day, int month, int year){
    // The arguments are used as values
    this.day = day;
    this.month = month;
    this.year = year;
  }

  // Parameterized constructor
  public Date(int day, int month, int year, String event){
    this(day, month, year); // calling the constructor
    this.event = event;
  }

  // A simple print function
  public void printDate(){
    System.out.println("Date: " + day + "/" + month + "/" + year + "  --> " + event);
  }

}

class Demo {

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

```
  public static void main(String args[]) {
    // Call the Date constructor to create its object;
    Date date = new Date(1, 1, 2019, "New Year"); // Object created with specified values! //
    date.printDate();
  }

}
```



The `this` keyword followed by parentheses means that another constructor in the same Java class is being called. At line **27** the first constructor in the class is being called.

This concludes our discussion on the basics of the classes in Java. The next section deals with the concept of data hiding, which plays a pivotal role in implementing efficient classes.