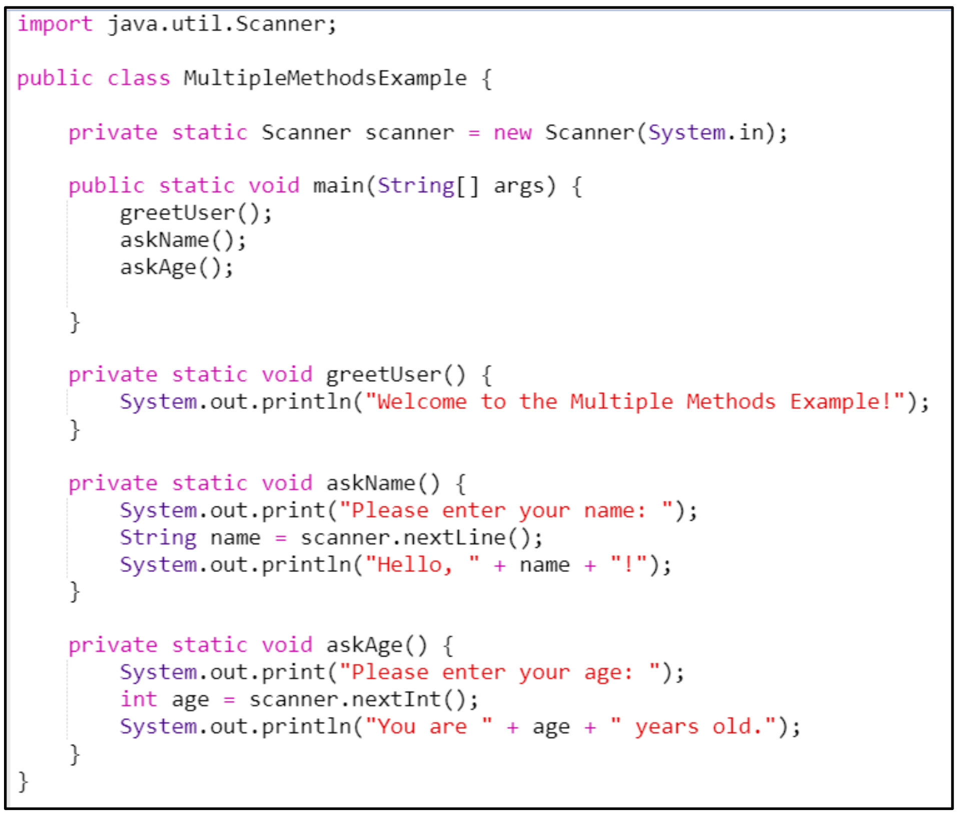
Java Method Overloading

Method Overloading

With **method overloading**, multiple methods can have the same name with different parameters:



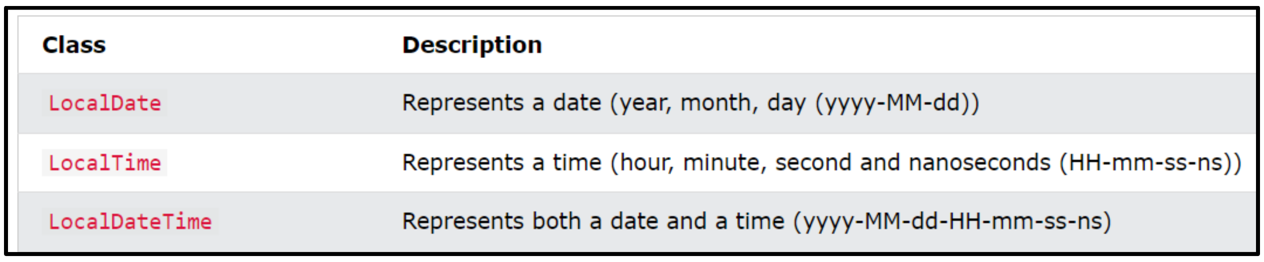
```
import java.util.Scanner;

public class MultipleMethodsExample {

    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        greetUser();
        askName();
        askAge();

    }

    private static void greetUser() {
        System.out.println("Welcome to the Multiple Methods Example!");
    }

    private static void askName() {
        System.out.print("Please enter your name: ");
        String name = scanner.nextLine();
        System.out.println("Hello, " + name + "!");
    }

    private static void askAge() {
        System.out.print("Please enter your age: ");
        int age = scanner.nextInt();
        System.out.println("You are " + age + " years old.");
    }
}
```

Advantage of method overloading

Method overloading *increases the readability of the program*.

Java Date and Time

Java does not have a built-in Date class, but we can import the java.time package to work with the date and time API. The package includes many date and time classes. For example:



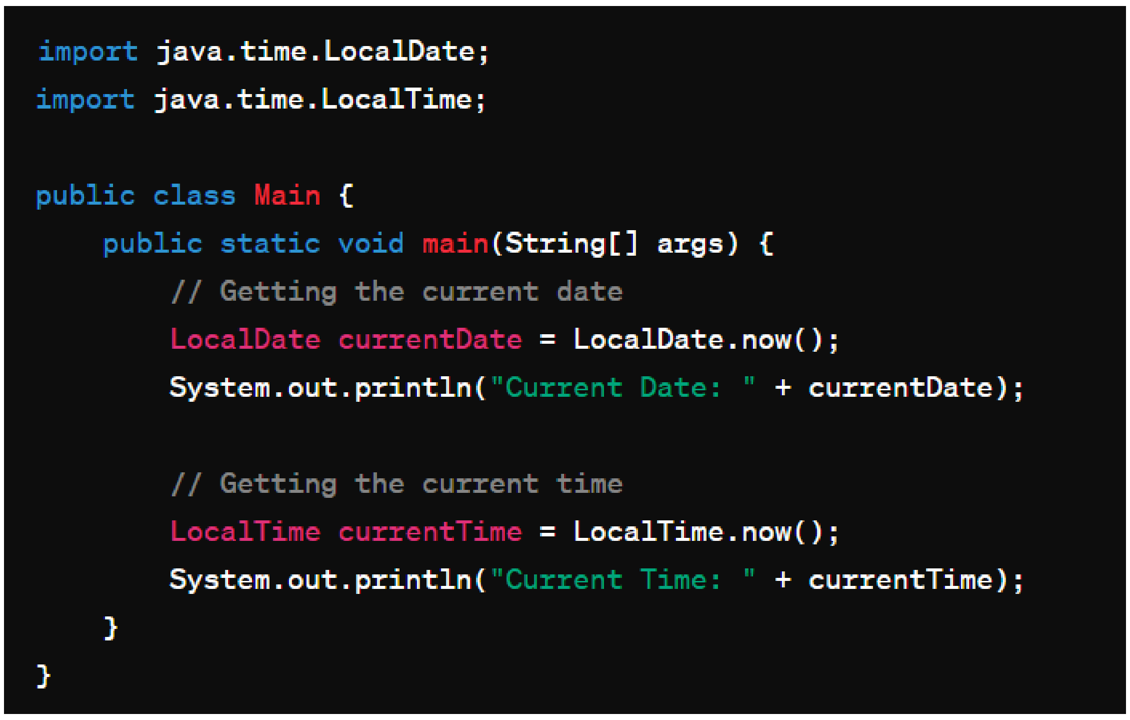
Class	Description
`LocalDate`	Represents a date (year, month, day (yyyy-MM-dd))
`LocalTime`	Represents a time (hour, minute, second and nanoseconds (HH-mm-ss-ns))
`LocalDateTime`	Represents both a date and a time (yyyy-MM-dd-HH-mm-ss-ns)

Display Current Date

To display the current date, import the java.time.LocalDate class, and use its now() method:

Display Current Time

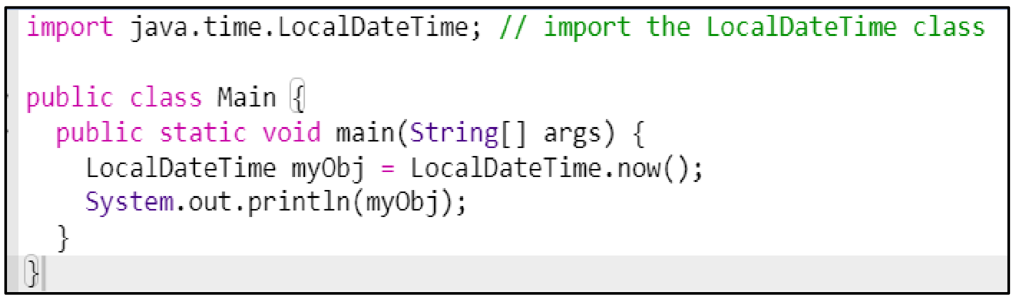
To display the current time (hour, minute, second, and nanoseconds), import the java.time.LocalTime class, and use its now() method:



```java
import java.time.LocalDate;
import java.time.LocalTime;

public class Main {
    public static void main(String[] args) {
        // Getting the current date
        LocalDate currentDate = LocalDate.now();
        System.out.println("Current Date: " + currentDate);

        // Getting the current time
        LocalTime currentTime = LocalTime.now();
        System.out.println("Current Time: " + currentTime);
    }
}
```

Display Current Date and Time

To display the current date and time, import the java.time.LocalDateTime class, and use its now() method:



```java
import java.time.LocalDateTime; // import the LocalDateTime class

public class Main {
  public static void main(String[] args) {
    LocalDateTime myObj = LocalDateTime.now();
    System.out.println(myObj);
  }
}
```