Java Encapsulation

Encapsulation

The meaning of **Encapsulation**, is to make sure that "sensitive" data is hidden from users. To achieve this, you must:

- declare class variables/attributes as private
- provide public **get** and **set** methods to access and update the value of a private variable

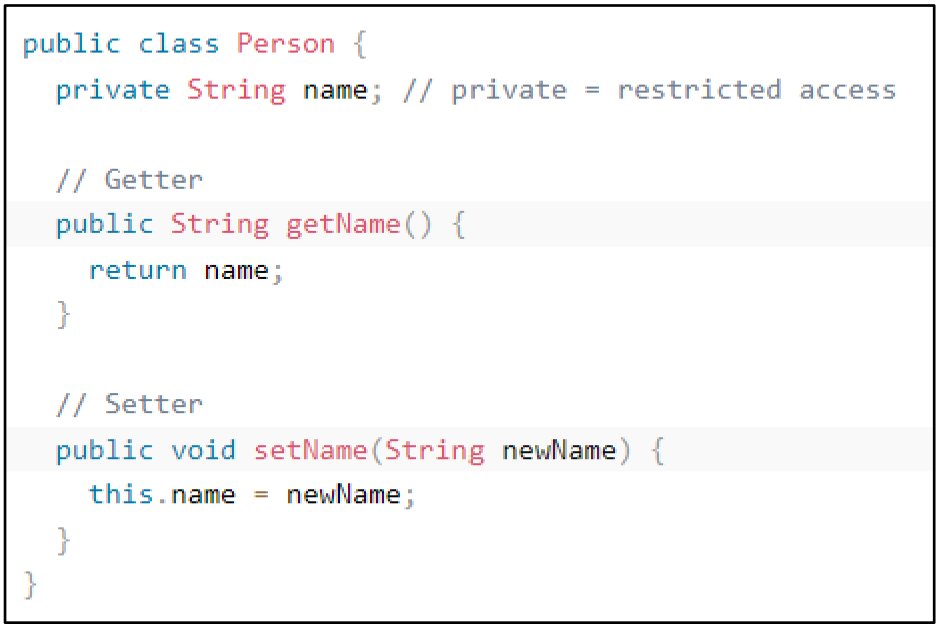
Data Hiding in Java is hiding the variables of a class from other classes. It can only be accessed through the method of their current class. It hides the implementation details from the users. But more than data hiding, it is meant for better management or grouping of related data.

Getters and Setters

You learned from the previous chapter that private variables can only be accessed within the same class (an outside class has no access to it). However, it is possible to access them if we provide public **get** and **set** methods.

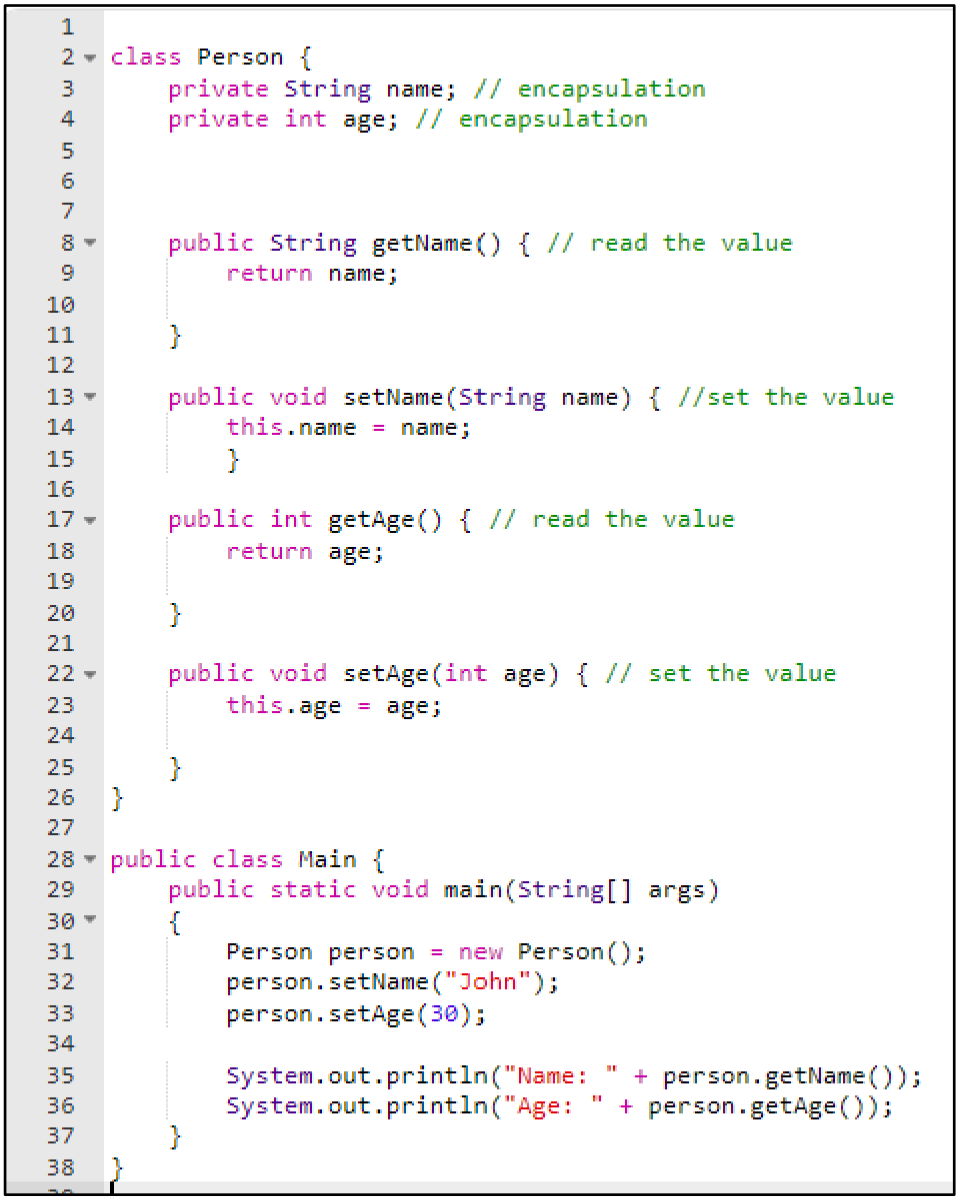
The **get** method returns the variable value, and the **set** method sets the value.

Syntax for both is that they start with either get or set, followed by the name of the variable, with the first letter in upper case:

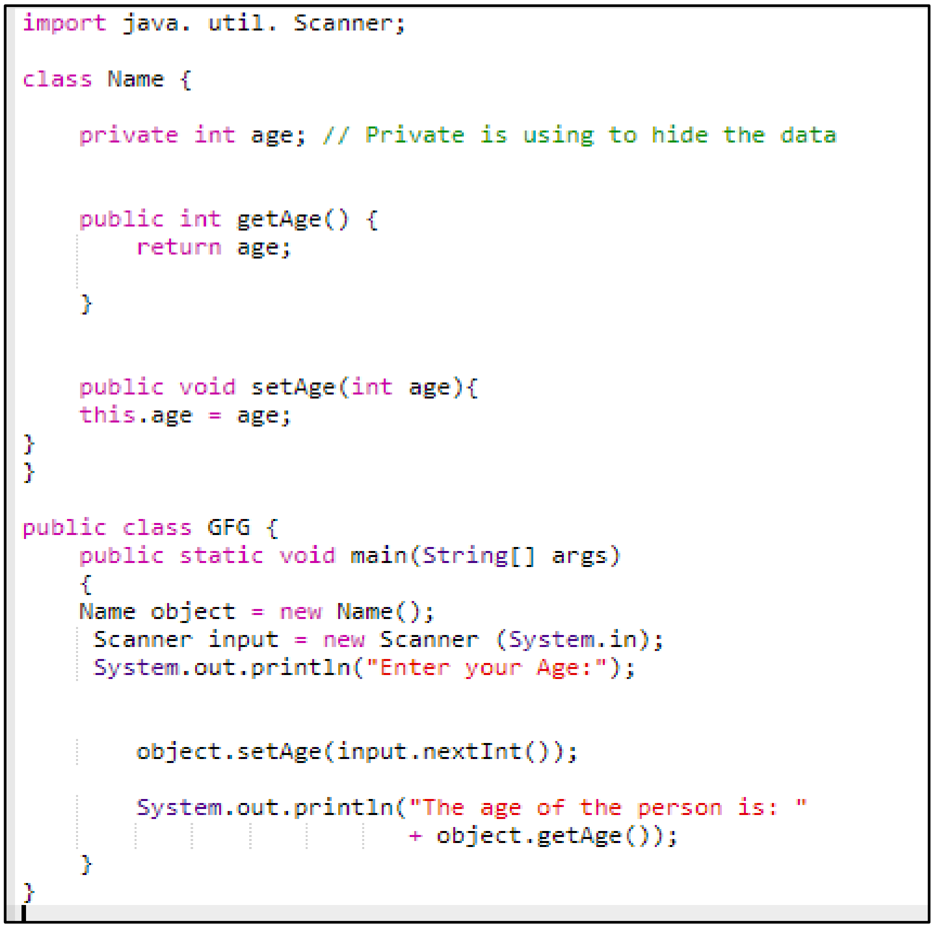


```
public class Person {
  private String name; // private = restricted access

  // Getter
  public String getName() {
    return name;
  }

  // Setter
  public void setName(String newName) {
    this.name = newName;
  }
}
```

EXAMPLE of ENCAPSULATION WITH AND WITHOUT SCANNER



```

class Person {
    private String name; // encapsulation
    private int age; // encapsulation



    public String getName() { // read the value
        return name;

    }

    public void setName(String name) { //set the value
        this.name = name;
        }

    public int getAge() { // read the value
        return age;

    }

    public void setAge(int age) { // set the value
        this.age = age;

    }
}

public class Main {
    public static void main(String[] args)
    {
        Person person = new Person();
        person.setName("John");
        person.setAge(30);

        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());
    }
}
```



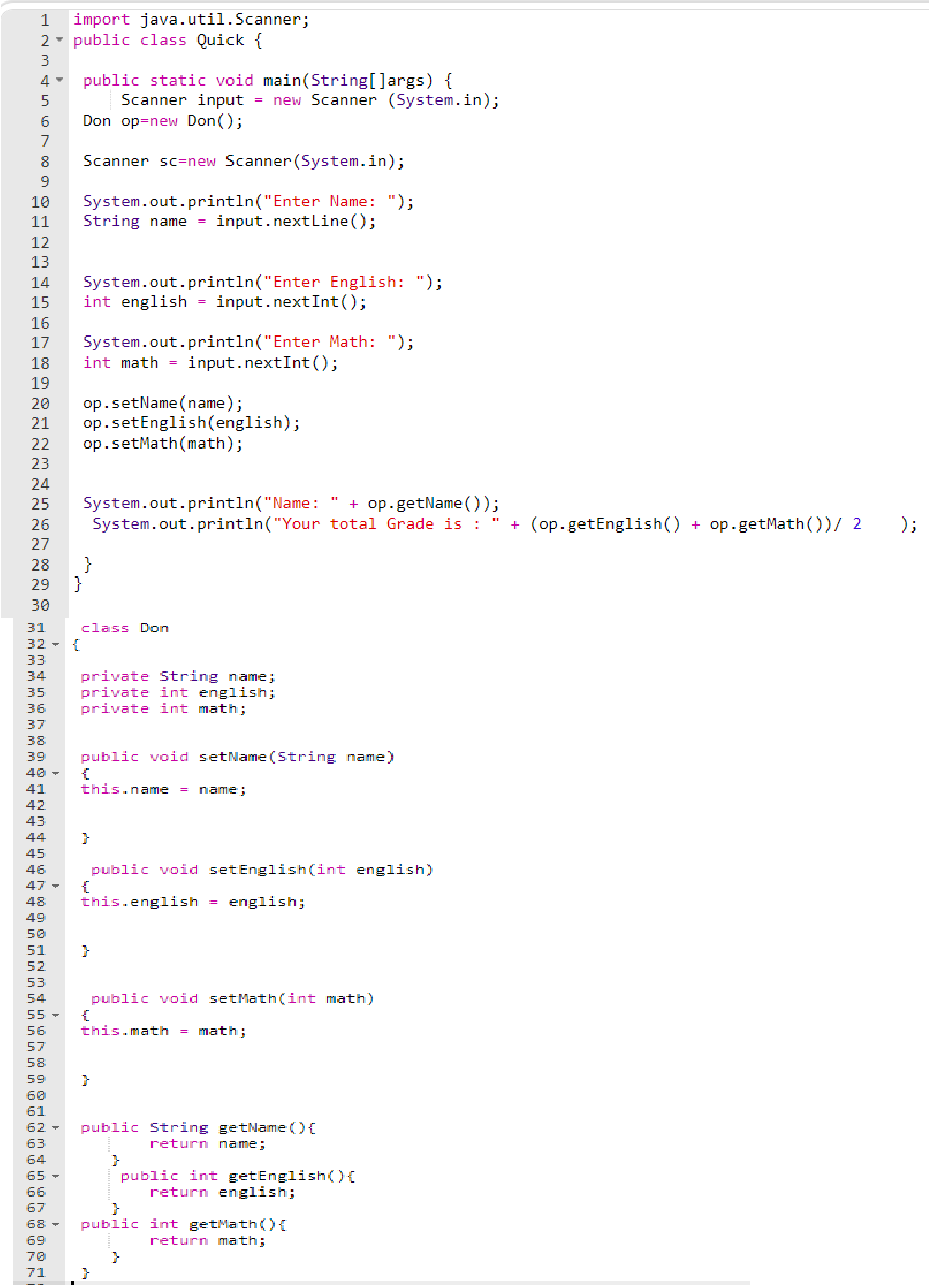
```
import java. util. Scanner;

class Name {

    private int age; // Private is using to hide the data


    public int getAge() {
        return age;

    }


    public void setAge(int age){
    this.age = age;
}
}

public class GFG {
    public static void main(String[] args)
    {
    Name object = new Name();
     Scanner input = new Scanner (System.in);
     System.out.println("Enter your Age:");


        object.setAge(input.nextInt());

        System.out.println("The age of the person is: "
                           + object.getAge());
    }
}
```

Why Encapsulation?

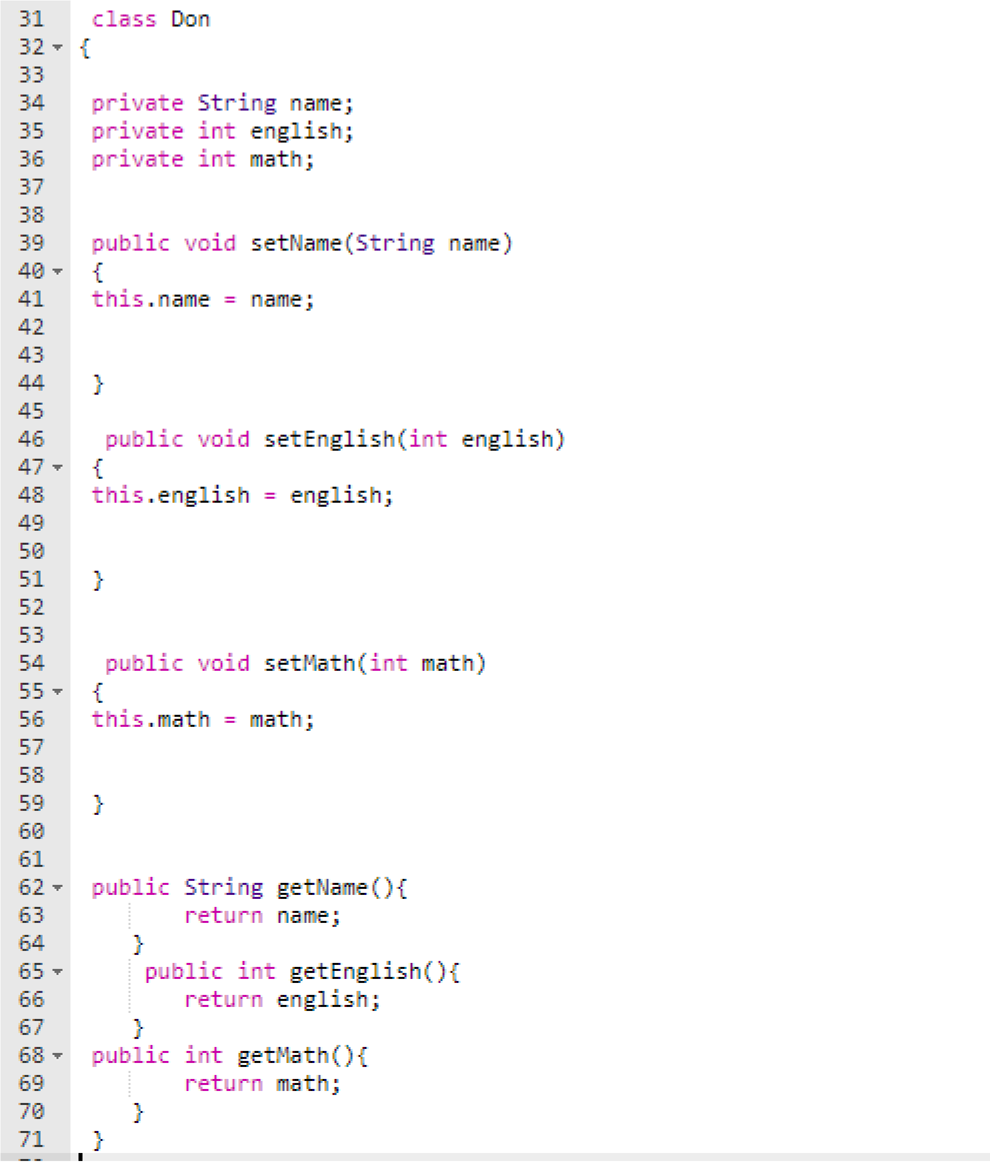
- Better control of class attributes and methods
- Class attributes can be made **read-only** (if you only use the get method), or **write-only** (if you only use the set method)
-
- Flexible: the programmer can change one part of the code without affecting other parts
- Increased security of data

Disadvantages of Encapsulation in Java:

1. Can lead to increased complexity, especially if not used properly.
2. Can make it more difficult to understand how the system works.
3. May limit the flexibility of the implementation.



```
import java.util.Scanner;
public class Quick {

 public static void main(String[]args) {
     Scanner input = new Scanner (System.in);
 Don op=new Don();

 Scanner sc=new Scanner(System.in);

 System.out.println("Enter Name: ");
 String name = input.nextLine();


 System.out.println("Enter English: ");
 int english = input.nextInt();

 System.out.println("Enter Math: ");
 int math = input.nextInt();

 op.setName(name);
 op.setEnglish(english);
 op.setMath(math);


 System.out.println("Name: " + op.getName());
  System.out.println("Your total Grade is : " + (op.getEnglish() + op.getMath())/ 2    );

 }
}

 class Don
{

 private String name;
 private int english;
 private int math;


 public void setName(String name)
 {
 this.name = name;


 }

  public void setEnglish(int english)
 {
 this.english = english;


 }


  public void setMath(int math)
 {
 this.math = math;


 }


 public String getName(){
        return name;
     }
     public int getEnglish(){
        return english;
     }
 public int getMath(){
        return math;
     }
 }
```



```
 class Don
{

 private String name;
 private int english;
 private int math;


 public void setName(String name)
 {
 this.name = name;


 }

  public void setEnglish(int english)
 {
 this.english = english;


 }


  public void setMath(int math)
 {
 this.math = math;


 }


 public String getName(){
        return name;
    }
     public int getEnglish(){
        return english;
    }
 public int getMath(){
        return math;
    }
 }
```