**Encapsulation:**

Encapsulation is the concept of wrapping the data (variables) and the code (methods) together as a single unit. It helps to protect the data by restricting direct access to it and allows access only through methods (getters and setters).

**Example:**

java

Copy code

```
class Person {
   // Private variables (data hiding)
   private String name;
   private int age;

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

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

   // Getter method
   public int getAge() {
      return age;
   }

   // Setter method
   public void setAge(int age) {
```

```
        this.age = age;
    }
}

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

        System.out.println(person.getName());  // Using getter
        System.out.println(person.getAge());   // Using getter
    }
}
```

**Explanation:**

- name and age are encapsulated inside the Person class.
- They cannot be directly accessed outside the class. Instead, we use getter and setter methods to get and set their values.

---

**Abstraction:**

Abstraction is the concept of hiding the complex implementation details and showing only the necessary features. It allows you to focus on what an object does, instead of how it does it.

**Example:**

java

Copy code

```
abstract class Animal {
    // Abstract method (no implementation)
    public abstract void sound();
}
```

```
class Dog extends Animal {
    // Providing implementation for the abstract method
    public void sound() {
        System.out.println("Bark");
    }
}


public class Main {
    public static void main(String[] args) {
        Animal animal = new Dog();
        animal.sound();  // Output: Bark
    }
}
```

**Explanation:**

- The Animal class is abstract, and it only defines the sound() method without implementing it.
- The Dog class provides the actual implementation of the sound() method.
- When we create an object of Dog and call the sound() method, we only see the result ("Bark"), without worrying about how the sound is made. This is abstraction.

---

**Key Differences:**

- **Encapsulation** hides the data and controls access through methods.
- **Abstraction** hides the implementation details and shows only the relevant functionality.

4o mini

in java