1. **What is Encapsulation in Java? Why is it called Data hiding?**
   Encapsulation is an object-oriented programming principle that bundles the data (variables) and the methods (functions) that operate on the data into a single unit, called a class. It is called data hiding because it restricts direct access to some of the object's components, which can only be accessed through public methods.

2. **What are the important features of Encapsulation?**
   The key features of encapsulation include:
   - **Data Hiding**: Protects the internal state of an object from unintended or harmful modifications.
   - **Modularity**: Each class is a self-contained module that can be developed, tested, and debugged independently.
   - **Flexibility**: Allows changes to the implementation without affecting other parts of the program.
   - **Maintainability**: Makes the code easier to maintain and modify.

3. **What are getter and setter methods in Java? Explain with an example.**
   Getter and setter methods are used to access and update the value of private variables. For example:

```
public class Person {
    private String name;

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

    // Setter method
    public void setName(String name) {
        this.name = name;
    }
}
public class Main {
    public static void main(String[] args) {

        Person person = new Person();

        person.setName("Ram");

        System.out.println(person.getName()); // Output: Ram

    }

}
```

4. **What is the use of the this keyword? Explain with an example.**
The this keyword is used to refer to the current instance of the class. It is often used to resolve naming conflicts between instance variables and parameters. For example:

```
public class Person {

    private String name;

    public Person(String name) {

        this.name = name; // 'this.name' refers to the instance variable, 'name'
refers to the parameter

    }

    public void display() {

        System.out.println("Name: " + this.name);

    }

}


public class Main {

    public static void main(String[] args) {

        Person person = new Person("Alice");

        person.display(); // Output: Name: Alice

    }

}
```

5. **What is the advantage of Encapsulation?**

Encapsulation provides several advantages:

- **Improved Security**: By hiding the internal state of objects, it prevents unauthorized access and modification.
- **Increased Flexibility**: Allows changes to the implementation without affecting other parts of the program.
- **Ease of Maintenance**: Makes the code easier to maintain and understand.
- **Reusability**: Encapsulated code can be reused across different parts of the program or in different programs.

6. **How to achieve encapsulation in Java? Give an example.**

Encapsulation is achieved by:

- Declaring the variables of a class as private.
- Providing public getter and setter methods to access and update the value of private variables.

For example

```
public class Employee {
    private int id;
    private String name;

    // Getter for id
    public int getId() {
        return id;
    }

    // Setter for id
    public void setId(int id) {
        this.id = id;
    }


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

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

public class Main {
    public static void main(String[] args) {
        Employee emp = new Employee();
        emp.setId(101);
        emp.setName("Ram");
        System.out.println("ID: " + emp.getId() + ", Name: " + emp.getName()); // Output: ID: 101, Name: Ram
    }
}
```