1. What is Encapsulation in Java? Why is it called Data Hiding:

Encapsulation in Java is the concept of wrapping data (variables) and code (methods) together into a single unit, typically a class. It is called data hiding because it restricts direct access to the internal state of the object, allowing controlled access through public methods.

2. What are the Important Features of Encapsulation:

- Data Hiding: Internal state of the object is hidden from the outside.

- Controlled Access: Provides controlled access to the variables via getter and setter methods.

- Modularity: Keeps the data safe from unintended access and misuse.

- Reusability: Enhances reusability and maintainability of the code.

3. What are Getter and Setter Methods in Java? Explain with an Example:

Getter and setter methods are used to access and modify the private variables of a class. The getter method returns the value of the variable, while the setter method sets or updates the value.

Example:

```
public class Person {
   private String name; // private variable
    // Getter method
   public String getName() {
      return name;
   }
    // Setter method
   public void setName(String name) {
      this.name = name;
   }

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

```
        Person person = new Person();
        person.setName("John"); // Set the name
        System.out.println("Name: " + person.getName()); // Get the name
    }
}
```

4. What is the Use of `this` Keyword? Explain with an Example:

The `this` keyword in Java is used to refer to the current object instance. It is often used to resolve conflicts between instance variables and parameters with the same name.

Example:

```
public class Employee {
    private String name;
    public Employee(String name) {
        this.name = name; // `this.name` refers to the instance variable
    }
    public void display() {
        System.out.println("Employee Name: " + this.name);
    }
    public static void main(String[] args) {
        Employee emp = new Employee("Alice");
        emp.display();
    }
}
```

5. What is the Advantage of Encapsulation:

- Improved Security: By hiding the internal implementation, it protects the object’s integrity.

- Flexibility: Internal implementation can be changed without affecting other parts of the code.
- Easy Maintenance: Encapsulation makes the code more modular and easier to maintain.
- Controlled Access: Only expose what is necessary, keeping the internal details hidden.

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 modify the private variables.

Example:

```
public class Car {
    private String model; // Private variable
    private int year;
    // Getter for model
    public String getModel() {
        return model;
    }
    // Setter for model
    public void setModel(String model) {
        this.model = model;
    }
    // Getter for year
    public int getYear() {
        return year;
    }
    // Setter for year
    public void setYear(int year) {
```

```
    this.year = year;
  }

  public static void main(String[] args) {
    Car car = new Car();
    car.setModel("Toyota");
    car.setYear(2020);
    System.out.println("Car Model: " + car.getModel());
    System.out.println("Car Year: " + car.getYear());
  }
}
```