Encapsulation

1.What is Encapsulation in Java? Why it is called data hiding?

Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.

2.What are the important features of Encapsulation?

Combine the data of our application and its manipulation at one place.
Encapsulation Allow the state of an object to be accessed and modified through behaviors.
Reduce the coupling of modules and increase the cohesion inside them.

3.What are getter and setter methods in Java. Explain with an example

For each instance variable, a getter method returns its value while a setter method sets or updates its value.

Example:

```
public class Vehicle {
  private String color;
  // Getter

  public String getColor() {
    return color;
  }
  // Setter

  public void setColor(String c) {
    this.color = c;
  }
}
```

4.What is the use of this keyword explain with an example

The this keyword refers to the current object in a method or constructor.
The most common use of the this keyword is to eliminate the confusion between class attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameter).

Example:

```
public class Main {
  int x;
  // Constructor with a parameter
  public Main(int x) {
    this.x = x;
  }
  // Call the constructor
  public static void main(String[] args) {
    Main myObj = new Main(5);
    System.out.println("Value of x = " + myObj.x);
  }
}
```

5.What is the advantage of Encapsulation

Following are the advantages of encapsulation in Java.

Protect Your Data

With encapsulation, you can keep your data and codes safe from external inheritance. For example, if any program runner tries to change the program, they can only interact with the getter and setter methods of the program. They will not have any idea to change any specific variable or data and hinder the running of the program resulting in high security.

Easy to Test code

The code which is encapsulated is simple to debug and easy to test for unit testing.

Flexible

The encapsulated code is cleaner, flexible, and easy to change as per our needs. It means we can change the code read-only or write-only by getter and setter methods.

For example, if you don't define the setter method in the class then the fields can be made read-only whereas if you don't define the getter method in the class then the fields can be made write-only.

Easy to Reuse

Encapsulation enables you to easily change the methods, reuse the code, and execute new requirements in your program.

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

- As in encapsulation, the data in a class is hidden from other classes using the data hiding concept which is achieved by making the members or methods of a class private, and the class is exposed to the end-user or the world without providing any details behind implementation using the abstraction concept, so it is also known as a combination of data-hiding and abstraction.

- Encapsulation can be achieved by Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

Example:

```
// fields to calculate area
class Area {

int length;
int breadth;

// constructor to initialize values
Area(int length, int breadth) {
	this.length = length;
	this.breadth = breadth;
}

// method to calculate area
public void getArea() {
	int area = length * breadth;
	System.out.println("Area: " + area);
}
}

class Main {
public static void main(String[] args) {

	Area rectangle = new Area(2, 16);
	rectangle.getArea();
}
}
```