

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

**=>**Binding of data and corresponding methods into a single unit is called "Encapsulation".

If any java class follows data hiding and abstraction then such class is referred as "Encapsulated class". Encapsulation = Data Hiding + abstraction.

Every data member inside the class should be declared as private, and to access this private data we need to have setter and getter methods.

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?**

**=>** **Data Hiding/Security:** it is a way of restricting the access of our data members by hiding the implementation details. Encapsulation also provides a way for data hiding. The user will have no idea about the inner implementation of the class. It will not be visible to the user how the class is storing values in the variables. The user will only know that we are passing the values to a setter method and variables are getting initialized with that value.

**Increased Flexibility:** We can make the variables of the class read-only or write-only depending on our requirement. If we wish to make the variables read-only then we have to omit the setter methods like setName(), setAge(), etc. from the above program or if we wish to make the variables write-only then we have to omit the get methods like getName(), getAge(), etc. from the above program

**Reusability:** Encapsulation also improves the re-usability and is easy to change with new requirements.

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

**=>** **Setter** methods are used to set the value to the instance variables of the class. Syntax for setter method

a. compulsory the method name should start with set. b. it should be public.   
c. return type should be void.  
d. compulsorily it should have some argument.

**Getter** methods are used to get the value from the instance variables of the class. Syntax for getter method

a. compulsory the method name should start with get.

b. it should be public.   
c. return type should not be void.  
d. compulsorily it should not have any argument.

***Example-***

class Student{  
 private int age;  
 private String name;

//Setters  
 public void setAge(int age)  
 {  
 this.age=age;  
  
 }  
 public int getAge()  
 {  
 return age;  
 }  
 public void setName(String name)  
 {  
 this.name="Navin";  
  
 }

//getters  
 public String getName()  
 {  
 return name;  
 }  
 public void show()  
 {  
 System.*out*.println(age+" "+name);  
 }  
}  
public class Demo8{  
 public static void main(String[] args){  
 Student obj=new Student();  
 Student obj1=new Student();  
 obj.setAge(18);  
 obj1.setAge(25);  
 obj.setName("Dipayan");  
 obj1.setName("kiran");  
 int stud1Age=obj.getAge();  
 String stud1Name=obj.getName();  
 System.*out*.println(stud1Age);  
 }  
}

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

**=>** If both local variable and instance variable have the same name inside the method then it would result in a name-clash and jvm will always give preference for local variable.

This approach is called the “Shadowing problem”. To avoid this problem we use **this** keyword  
**this** keyword would always point to current object, and this variable would hold the address the active object present in the heap memory.

***Example-***class Student7{  
 private int age;  
 private String name;  
  
 public void setData1(int a)  
 {  
 this.age=a;  
  
 }  
 public void setData2(String name) {  
 this.name=name;  
  
 }  
 public void show()  
 {  
 System.*out*.println(age+" "+name);  
 }  
}  
public class Demo7{  
 public static void main(String[] args){  
 Student7 obj=new Student7();  
 Student7 obj1=new Student7();  
 obj.setData1(18);  
 obj1.setData1(25);  
 obj.setData2("Dipayan");  
 obj1.setData2("kiran");  
 obj.show();  
 obj1.show();  
 }  
}

* As noticed in the above program, the variables name,id,address are local variables and these values should be assigned to instance variables of student class\
* Inside the method the jvm will always give preference only for local variables, this problem is termed as “Shadowing”\
* To resolve this problem we need to use , “this” keyword.

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

**=>*Advantages of Encapsulation***   
a. We can achieve security.   
b. Enhancement becomes easy.   
c. Maintainability and modularisation becomes easy.   
d. It provides flexibility to the user to use the system very easily.

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

### =>*Advantages of Encapsulation in Java:*

1. Improves security of an object’s internal state by hiding it from the outside world.
2. Increases modularity and maintainability by making it easier to change the implementation without affecting other parts of the code.
3. Enables data abstraction, allowing objects to be treated as a single unit.
4. Allows for easy addition of new methods and fields without affecting the existing code.
5. Supports the object-oriented principle of information hiding, making it easier to change the implementation without affecting the rest of the code.