Encapsulation

Assignment

Question 1- What is Encapsulation in Java? Why is it called Data hiding?

Ans-

Encapsulation in Java is a way of combining data and behavior within an object and restricting external access to the object’s internal data. It is called “data hiding” because it involves restricting access to the object’s data, ensuring that it is only accessed and modified through a specific method defined within the object. This improves the security and reliability of code by ensuring that data is only modified in a controlled manner.

Question 2- What are the important features of Encapsulation?

Ans-

Encapsulation is an important concept in Java that helps ensure data security and maintainability of code. The key features of Encapsulation in are:

1. Data Hiding: Encapsulation hides the data or implementation details of a class from the outside world, thus preventing direct access to data from outside of the class.
2. Access Modifiers: Java provides access modifiers (public, private, protected) to restrict the visibility of class members. These access modifiers help to enforce encapsulation and prevent unintended modifications.
3. Getters and Setter: Encapsulation is achieved by providing access to class members through public methods like getter and setter. These methods provide controlled access to the class members while also allowing the class to enforce any constraints or rules.
4. Class Abstraction: Encapsulation helps in achieving class abstraction, which means that the complex implementation details of a class are hidden and only a simplified interface is exposed to the outside world.

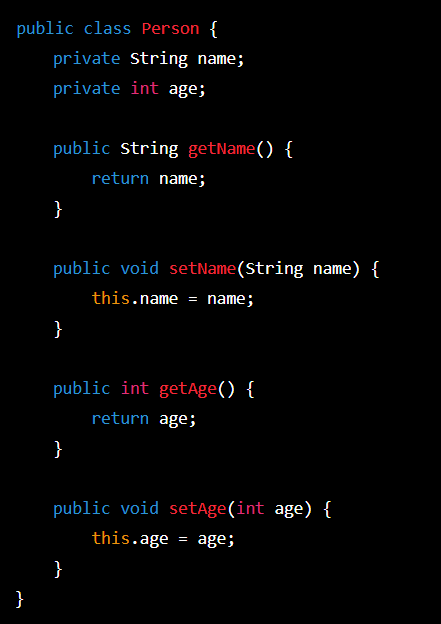
Overall, Encapsulation in Java provides a mechanism to protect the internal data and implementation details of a class, and also ensures that the class can maintain its internal consistency and integrity.

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

Ans-

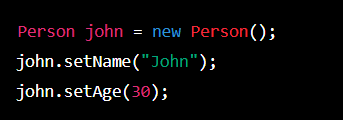
Getter and setter methods, also known as accessor and mutator methods, are a pair of methods used to access and modify the value of private variables in a class. They are an essential part of encapsulation, which is a fundamental principle of object-oriented programming that protects the internal state of an object from outside interference.

Here’s an example of getter and setter method:

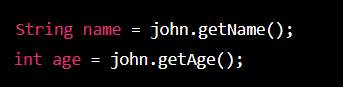


In this example, the ‘person’ class has two private instance variables, ‘name’ and ‘age’. The ‘getName()’and’ getAge()’ methods are getter methods that allow the outside world to access the values of these private variables. The ‘setName()’ and ‘setAge()’ methods are setter methods that allow the outside world to modify the values of these private variables.

For example, if we create a ‘person’ object named ‘john’ and ant to set his name to “john” and his age to 30, we can do so using the setter methods:



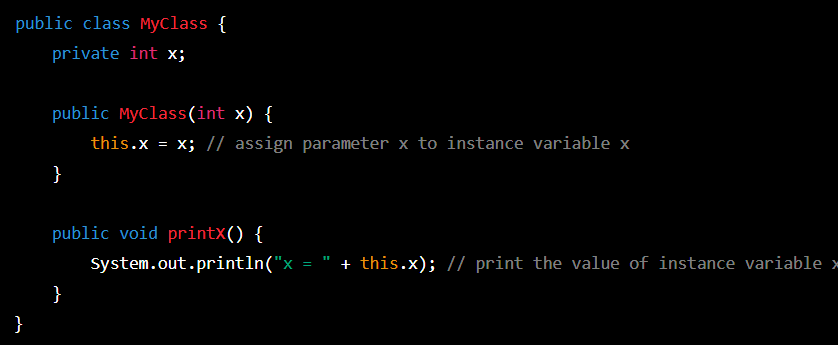
And if we want to retrieve john’s name and age, we can do so using the gitter methods:



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

Ans-

The ‘this’ keyword in Java is used to refer to the current object instance of a class. It can be used to access instance variables, methods, or constructors of the current object, and to avoid naming conflicts with local variables or parameters. For example:



Here,’this.x’ is used to refer to the instance variable ‘x’ of the current object, and to assign the value of the parameter ‘x’ to it in the constructor. In the ‘printx’ method,’this.x’ is used to print the value of the instance variable ‘x’.

Question 5- What is the advantage of Encapsulation?

Ans-

Encapsulation is a fundamental concept in object-oriented programming that allows the hiding of implementation details of a class from the outside world, and providing a controlled access to it through public methods. This provides the following advantages:

1. Data hiding: The implementation details of a class can be hidden from the outside world, preventing unauthorized access and modification of the data.
2. Code reusability: Encapsulation facilitates code reusability as the encapsulated data can be used in other parts of the program without exposing its internal implementation.
3. Improved maintainability: Encapsulation allows the implementation of a class to be modified without affecting the external clients that use it, thus improving the maintainability of the code.
4. Better control: Encapsulation provides better control over the data and operations of a class, making it easier to debug and maintain the code.
5. Security: Encapsulation ensures that the data is accessed only through controlled methods, which can help prevent unauthorized access or malicious attacks.

Question 6- How to achieve encapsulation in Java ? Given an example.

Ans -

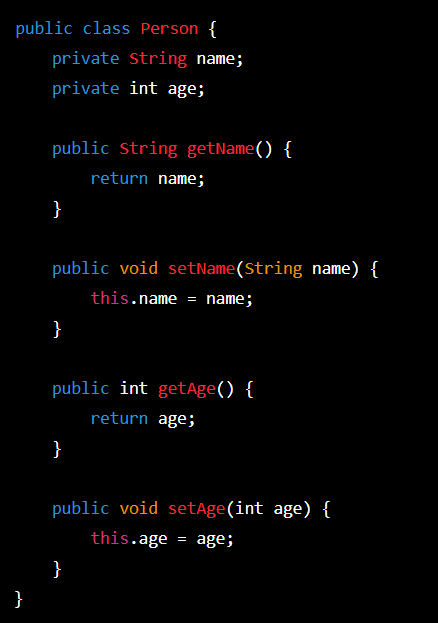
Encapsulation is a fundamental principle of object-oriented programming that involves bundling data and methods that operate on that data within a single unit, called a class.

Encapsulation helps to ensure that the implementation details of a class are hidden from the

Outside world, making it easier to manage and modify code.

In Java, encapsulation can be achieved by declaring the instance variables of a class as private and providing public getter and setter methods to access and modify those variables. This way,the internal state of the object is protected from direct external access, and can only be accessed through the public methods of the class.

Here’s a simple example:



In this example, the ‘name’ and ‘age’ instance variables are declared as private, which means that they can be accessed within the ‘person’ class. However , public getter and setter methods are provided(‘getName()’, ‘setName()’, ‘getAge()’, and ‘setAge()’) to allow external code to read and modify the values of these variables. This way, the implementation details of the ‘person’ class are hidden from the outside world, and the internal state of the ‘person’ object is protected.