CORE JAVA

MANUAL V8.3

MODULE CODE:

ANUDIP FOUNDATION

ICONS AND THEIR MEANING
HINTS:
Get ready for helpful insites on difficult topics and questions.
STUDENTS:
This icon symbolize important instrcutions and guides for the students.
TEACHERS/TRAINERS:
This icon symbolize important instrcutions and guides for the trainers.

Module 2: Object Oriented Programming and Package

Chapter 5

Objective: After completing this lesson you will be able to : * Understand the concept of Java encapsulation * Gain an understanding of abstraction in Java	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 60 minutes	**Practical Duration:** 60 minutes
Total Duration: 120 minutes	

Chapter 5

5.1 Encapsulation

In Java, encapsulation refers to a process of wrapping variables and methods as a single unit. Class variables are hidden away from other classes, and they can only be accessed through their current class methods. The mechanism of encapsulation is also referred to as 'data hiding' as it hides data from other classes.

Encapsulation is a key concept of object oriented programming, along with abstraction, polymorphism and inheritance.

Encapsulation in Java can be achieved by –

- Declaring all variables within a class as private
- Enabling setter and getter methods for viewing and modifying variable values.

The advantages of encapsulation are –

i) Programmers can make Java class fields write-only or read-only

ii) Classes gain full control over the content of fields

iii) Specific sections of code can be hidden with the process

iv) Testing becomes easier and more efficient

Example of encapsulation in Java –

```
class EncapsulationDemo{

   private int Aadhar;

   private String empName;

   private int empAge;
```

```
 public int getEmpAadhar(){

        return aadhar;

    }

    public String getEmpName(){

        return empName;

    }

     public int getEmpAge(){

        return empAge;

    }

    public void setEmpAge(int newValue){

        empAge = newValue;

    }

    public void setEmpName(String newValue){

        empName = newValue;

    }

    public void setEmpAadhar(int newValue){

        aadhar = newValue;

    }

}

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

        EncapsulationDemo obj = new EncapsulationDemo();

        obj.setEmpName('Ram');

        obj.setEmpAge(28);

        obj.setEmpAadhar(1111 2222 3333);

        System.out.println('Employee Name: ' + obj.getEmpName());

        System.out.println('Employee Aadhar: ' + obj.getEmpAadhar());

        System.out.println('Employee Age: ' + obj.getEmpAge());

   }

}
```

Output:

```
Employee Name: Ram

Employee Aadhar: 1111 2222 3333

Employee Age: 28
```

5.2 Abstraction

Abstraction in Java is the procedure used for hiding some details and exhibiting others to users. An abstraction process helps in identifying only the requisite object properties while avoiding unnecessary details.

Abstraction can be achieved through –

* interfaces – enables full abstraction

* abstract classes – enables partial abstraction

Advantages of Abstraction

1. Minimizes the complexity of observing things
2. Increases reusability
3. Code duplication is minimized
4. Raises application security by hiding details
5. Multiple related classes can be grouped as siblings

Practical (60 minutes)

See the example programme for Java encapsulation below. Write the same programme to display empName, empAge and empAadhar for the values Ramu, 26, 1222 1222 1222 respectively. Repeat the same programme for the values Ganesh, 22, 4444 4444 4444. Show the resulting outputs of both programs.

```
class EncapsulationDemo{

   private int Aadhar;

   private String empName;

   private int empAge;



 public int getEmpAadhar(){

      return aadhar;

   }

   public String getEmpName(){

      return empName;

   }

    public int getEmpAge(){
      return empAge;

   }
   public void setEmpAge(int newValue){
      empAge = newValue;
   }
   public void setEmpName(String newValue){
      empName = newValue;
   }
   public void setEmpAadhar(int newValue){
      aadhar = newValue;
   }
}
public class EncapsTest{

  public static void main(String args[]){
       EncapsulationDemo obj = new EncapsulationDemo();
       obj.setEmpName(‘Ram‘);
       obj.setEmpAge(28);
       obj.setEmpAadhar(1111 2222 3333);
       System.out.println(‘Employee Name: ‘ + obj.getEmpName());
       System.out.println(‘Employee Aadhar: ‘ + obj.getEmpAadhar());
```

```
    System.out.println(‘Employee Age: ‘ + obj.getEmpAge( ));

   }

}
```

Instructions: The progress of students will be assessed with the exercises mentioned below.

MCQ

1. Methods and ________ are wrapped as a single unit through encapsulation.

a) mechanisms

b) variants

c) variables

d) None of the mentioned

2. Encapsulation can be used to hide class variables from other _______.

a) modifiers

b) classes

c) subclasses

d) None of the mentioned

3. What is encapsulation also referred to as?

a) data hiding

b) data handling

c) data segregation

d) data classification

4. One of the methods of achieving encapsulation is declaring class variables as __________.

a) protected

b) private

c) public

d) None of the mentioned

5. Programmers can use encapsulation to make Java class fields read-only or _____ only.

a) rotation

b) copy

c) write

d) None of the mentioned

6. Encapsulation can be used to help classes gain full control over the content of ________.

a) fields

b) domains

c) brackets

d) None of the mentioned

7. Encapsulation makes _______ code easier.

a) rejecting

b) testing

c) re-forming

d) None of the mentioned

8. __________ is the process of hiding some details and showing others.

a) abstraction

b) selection

c) dissemination

d) None of the mentioned

9. Full abstraction can be achieved by using ___________.

a) properties

b) interfaces

c) abstract classes

d) None of the mentioned

10. The process of abstraction raises ___________.

a) redundancy

b) referability

c) reusability

d) None of the mentioned