Assignment - 6

Object-Oriented Programming in Java

Name: Kamithkar Vinod
Course: PG DAC AUGUST 2025
Form No: 250500480
Date: 16-09-2025

Problem 1: Use Setter, Getter

Task: Define a class of type Student that has rollno, name and age as private data members. Define SetData() and GetData() as public member functions with appropriate functionality. Write a program that declares 2 student objects, initializes the first at run-time and second by reading from console, and then displays both student's data.

Code:

```java
import java.util.Scanner;
class Student {
    int rno;
    String name;
    int age;

    // setter
    void setStudent(int rno, String name, int age) {
        this.rno = rno;
        this.name = name;
        this.age = age;
    }

    // getter accessor
    void disp() {
        System.out.println("Roll No: "+rno+" Name: "+name+" Age:
            "+age);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        Student s1 = new Student();
        s1.setStudent(1, "Anushka", 40);

```

```
        System.out.print("\nEnter Roll No: ");
        int r = scanner.nextInt();

        scanner.nextLine();
        System.out.print("\nEnter Student Name: ");
        String n = scanner.nextLine();

        System.out.print("\nEnter your Age: ");
        int a = scanner.nextInt();

        Student s2 = new Student();
        s2.setStudent(r, n, a);

        System.out.println("Displaying Student 2 Details: ");
        s2.disp();

        System.out.println("Displaying Student 1 Details: ");
        s1.disp();
    }
}
```

Output: —

```
D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\
6_Assignment>javac Student.java

D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\
6_Assignment>java Student

Enter Roll No: 2

Enter Student Name: Shyam

Enter your Age: 26
Displaying Student 2 Details:
Roll No: 2 Name: Shyam Age: 26
Displaying Student 1 Details:
Roll No: 1 Name: Anushka Age: 40
```

Problem 2: Use Default Constructor

Task: Create a class Person with attributes name, age and country. Implement methods to set and get these attributes. Create an object of this class, set its attributes, and print out the details.

Code: —

```
import java.util.Scanner;
class Person {
    String name;
    int age;
    static String country = "India";

    Person() {
```

```
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Person Details name and age: ");
        System.out.print("Enter your name: ");
        name = scanner.nextLine();
        System.out.print("\nEnter your age: ");
        age = scanner.nextInt();
    }

    void details() {
        System.out.println("Details: "+name + " "+ age + " " + country);
    }

    public static void main(String[] args) {
        Person p1 = new Person();
        p1.details();
    }
}
```

Output: —

```
D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\6_Assignment>javac Person.java

D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\6_Assignment>java Person
Enter Person Details name and age:
Enter your name: Vinod

Enter your age: 26
Details: Vinod 26 India
```

Problem 3: Constructor Overloading, default, parameterised

Task: Constructor Overloading: Extend the Person class from the previous problem and add multiple constructors (default, parameterized, etc.) to initialize the attributes. Also, include a method to display the details.

Code: —

```
import java.util.Scanner;
class _2Person {
    String name;
    int age;
    static String country = "India";

    _2Person() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Person Details name and age: ");
        System.out.print("Enter your name: ");
        name = scanner.nextLine();
```

```
12          System.out.print("\nEnter your age: ");
13          age = scanner.nextInt();
14      }
15
16      _2Person(String name, int age) {
17          this.name = name;
18          this.age = age;
19      }
20
21      void details() {
22          System.out.println("Details: "+name + " "+ age + " " +
                country);
23      }
24
25      public static void main(String[] args) {
26          _2Person p1 = new _2Person();
27          p1.details();
28
29          _2Person p2 = new _2Person("Dhoni", 45);
30          p2.details();
31      }
32  }
```

Output: —

```
D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\
6_Assignment>javac _2Person.java

D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\
6_Assignment>java _2Person
Enter Person Details name and age:
Enter your name: Vinod

Enter your age: 26
Details: Vinod 26 India
Details: Dhoni 45 India
```

Problem 4: this keyword

Task: Using this: Modify the Person class to include a method that displays the name and age of the object. Use this keyword to differentiate between class variables and method parameters. Implement a method to compare two Person objects based on their age.

Code: —

```
1   class Person {
2       private String name;
3       private int age;
4
5       public Person (String name, int age) {
6           // this refer's to instance variable
7           this.name = name;
```

```
8           this.age = age;
9       }
10
11      public void display() {
12          System.out.println("Name: " + name + " Age: " + age);
13      }
14
15      public void compareAge (Person other) {
16          if (this.age > other.age) {
17              System.out.println(this.name + " is older than " +
                    other.name);
18          }
19          else if (this.age < other.age) {
20              System.out.println(this.name + " is younger than " +
                    other.name);
21          }
22          else {
23              System.out.println(this.name + " and " + other.name +
                     " are of same age");
24          }
25
26      }
27
28      public static void main(String[] args) {
29          Person p1 = new Person("Vinod", 26);
30          Person p2 = new Person("Sony", 23);
31
32          p1.display();
33          p2.display();
34
35          p1.compareAge(p2);
36
37      }
38  }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_Practice\3_OOP_Java\1_Assignments\6_Assignment>
java Person
Name: Vinod Age: 26
Name: Sony Age: 23
Vinod is older than Sony
```

Problem 5: Static Variable, static block

Task: Static Variable: Create a class BankAccount with accno, accType, Balance and static variable interestRate. Initialize it using a static block. Implement methods to deposit and withdraw funds. Create objects and display details.

Code:

```
class BankAccount {
    private int accno;
    private String accType;
    private double balance;
    private static double interestRate;

    static {
        interestRate = 4.5;
        System.out.println("Staic Block Executed: Interet Rate
           Initialized to "+interestRate);
    }

    public BankAccount(int accno, String accType, double balance)
        {
        this.accno = accno;
        this.accType = accType;
        this.balance = balance;
    }

    // deposit method
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount + " | New
               Balance: " + balance);
        }
        else {
            System.out.println("Deposit Amount should be > than
               zero.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + " | New
               Balance: " + balance);
        }
        else {
            System.out.println("Invalid Withdrawl or Insufficient
                Balance");
        }
    }

    // withdraw method

    public void display(){
        System.out.println("Account No: " + accno);
        System.out.println("Account Type: " + accType);
        System.out.println("Balance: " + balance);
```

```
45          System.out.println("Interest Rate: " + interestRate);
46          System.out.println("----------------------------");
47      }
48
49
50      public static void main(String[] args) {
51          BankAccount acc1 = new BankAccount(101, "Savings", 4300);
52          BankAccount acc2 = new BankAccount(102, "Business",
                10000);
53
54          acc1.display();
55          acc2.display();
56
57          acc1.deposit(2000);
58          acc1.withdraw(1500);
59
60          acc2.deposit(5000);
61          acc2.withdraw(12000);
62
63          acc1.display();
64          acc2.display();
65
66      }
67  }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_Practice\3_OOP_Java\1_Assignments\6_Assignment>java BankAccount
Staic Block Executed: Interet Rate Initialized to 4.5
Account No: 101
Account Type: Savings
Balance: 4300.0
Interest Rate: 4.5
----------------------------
Account No: 102
Account Type: Business
Balance: 10000.0
Interest Rate: 4.5
----------------------------
Deposited: 2000.0 | New Balance: 6300.0
Withdrawn: 1500.0 | New Balance: 4800.0
Deposited: 5000.0 | New Balance: 15000.0
Withdrawn: 12000.0 | New Balance: 3000.0
Account No: 101
Account Type: Savings
Balance: 4800.0
Interest Rate: 4.5
----------------------------
Account No: 102
Account Type: Business
Balance: 3000.0
Interest Rate: 4.5
----------------------------
```

Problem 6: Static method

Task: Static Method: Add a static method to the BankAccount class from the previous problem to calculate interest based on a given balance and interest rate. Also, implement a method to display the account details including balance and interest earned.

Code:

```
class BankAccount {
    private int accno;
    private String accType;
    private double balance;
    private static double interestRate;

    static {
        interestRate = 4.5;
        System.out.println("Staic Block Executed: Interet Rate
           Initialized to "+interestRate);
    }

    public BankAccount(int accno, String accType, double balance)
        {
        this.accno = accno;
        this.accType = accType;
        this.balance = balance;
    }

    // deposit method
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount + " | New
               Balance: " + balance);
        }
        else {
            System.out.println("Deposit Amount should be > than
               zero.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + " | New
               Balance: " + balance);
        }
        else {
            System.out.println("Invalid Withdrawl or Insufficient
                Balance");
        }
    }

    // withdraw method

    public void display(){
        double interest = calculateInterest(this.balance);
        System.out.println("Account No: " + accno);
        System.out.println("Account Type: " + accType);
```

```
45          System.out.println("Balance: " + balance);
46          System.out.println("Interest Earned: " + interest);
47          System.out.println("Interest Rate: " + interestRate + "%"
               );
48          System.out.println("-----------------------------");
49      }
50
51      public static double calculateInterest (double balance) {
52          return (balance * interestRate) / 100;
53      }
54
55
56      public static void main(String[] args) {
57          BankAccount acc1 = new BankAccount(101, "Savings", 4300);
58          BankAccount acc2 = new BankAccount(102, "Business",
               10000);
59
60          acc1.display();
61          acc2.display();
62
63          acc1.deposit(2000);
64          acc1.withdraw(1500);
65
66          acc2.deposit(5000);
67          acc2.withdraw(12000);
68
69          acc1.display();
70          acc2.display();
71
72      }
73  }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_Practice\3_OOP_Java\1_Assignments\6_Assignment>java BankAccount
Staic Block Executed: Interet Rate Initialized to 4.5
Account No: 101
Account Type: Savings
Balance: 4300.0
Interest Earned: 193.5
Interest Rate: 4.5%
---------------------------
Account No: 102
Account Type: Business
Balance: 10000.0
Interest Earned: 450.0
Interest Rate: 4.5%
---------------------------
Deposited: 2000.0 | New Balance: 6300.0
Withdrawn: 1500.0 | New Balance: 4800.0
Deposited: 5000.0 | New Balance: 15000.0
Withdrawn: 12000.0 | New Balance: 3000.0
Account No: 101
Account Type: Savings
Balance: 4800.0
Interest Earned: 216.0
Interest Rate: 4.5%
---------------------------
Account No: 102
Account Type: Business
Balance: 3000.0
Interest Earned: 135.0
Interest Rate: 4.5%
---------------------------
```

Problem 7: this in constructor

Task: Using this in Constructors: Create a class Rectangle with attributes length and width. Implement a parameterized constructor that initializes these attributes. Use this to differentiate between class variables and constructor parameters. Include methods to calculate the area and perimeter.

Code:

```java
class Rectangle {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    public double calculateArea () {
        return this.length * this.width;
    }

    public double calculatePerimeter () {
        return 2 * (this.length + this.width);
    }

     public void display() {
        System.out.println("Length: " + this.length);
        System.out.println("Width: " + this.width);
        System.out.println("Area: " + calculateArea());
        System.out.println("Perimeter: " + calculatePerimeter());
        System.out.println("---------------------------");
```

```
    }

    public static void main(String[] args) {
        Rectangle r1 = new Rectangle(10, 20);
        Rectangle r2 = new Rectangle(15, 20);

        r1.display();
        r2.display();
    }
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_Practice\3_OOP_Java\1_Assignments\6_Assignment>
javac Rectangle.java

V:\CDAC\CDAC_PG_DAC_Practice\3_OOP_Java\1_Assignments\6_Assignment>
java Rectangle
Length: 10.0
Width: 20.0
Area: 200.0
Perimeter: 60.0
-------------------------
Length: 15.0
Width: 20.0
Area: 300.0
Perimeter: 70.0
-------------------------
```

Problem 8: Class and Methods

Task: Class and methods: Create a class Calculator with relevant data members and a constructor. Implement methods for basic arithmetic operations (addition, subtraction, multiplication, division, modulus) and demonstrate their usage.

Code: —

```
public class Calculator {
    private int a;
    private int b;

    public Calculator(int a, int b){
        this.a = a;
        this.b = b;
    }

    public void addition (){
        double n = a + b;
        System.out.println("Addition of " + a + " and " + b + "
            is: " + n);
    }

    public void substraction (){
        double n = a - b;
```

```
17          System.out.println("Substraction of " + a + " and " + b +
                " is: " + n);
18      }
19
20      public void multiplication (){
21          double n = a * b;
22          System.out.println("Multiplication of " + a + " and " + b
                + " is: " + n);
23      }
24
25      public void division (){
26          double n = a / b;
27          System.out.println("Division of " + a + " and " + b + "
                is: " + n);
28      }
29
30      public void modulus (){
31          double n = a % b;
32          System.out.println("Modulus of " + a + " and " + b + " is
                : " + n);
33      }
34  }
35
36  ////////////////////////////////////////////////////
37
38  import java.util.Scanner;
39
40  class Main {
41      public static void main(String[] args) {
42
43          Scanner sc = new Scanner(System.in);
44          System.out.print("Enter First Number: ");
45          int n1 = sc.nextInt();
46          System.out.print("\nEnter Second Number: ");
47          int n2 = sc.nextInt();
48          Calculator c = new Calculator(n1, n2);
49
50          System.out.println();
51          c.addition();
52          c.substraction();
53          c.division();
54          c.modulus();
55          c.multiplication();
56      }
57  }
```

Output: —

```
D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\
6_Assignment>java Main
Enter First Number: 7

Enter Second Number: 3
Addition of 7 and 3 is: 10.0
Substraction of 7 and 3 is: 4.0
Division of 7 and 3 is: 2.0
Modulus of 7 and 3 is: 1.0
Multiplication of 7 and 3 is: 21.0
```

Problem 9: Composition and Aggregation

Task: Composition and Aggregation: Create a class Address with attributes street, city, and state. Then create a class Person with attributes name and an Address object. Demonstrate how to use com Write a Java class representing a Student. Encapsulate the student's name, age, and grade point average (GPA) with private access modifiers. Provide getter and setter methods to access and modify these attributes position to model the relationship between a person and their address

Code:

```java
public class Address {
    private String street;
    private String city;
    private String state;

    public Address (String street, String city, String state) {
        this.street = street;
        this.city = city;
        this.state = state;
    }

    @Override
    public String toString() {
        return "Address {" +
            "street= " + street +
            " City= " + city +
            " State= " + state;
    }
}

////////////////////////////

public class Person {
    private String name;
    private Address address;

    public Person (String name, Address address) {
        this.name = name;
        this.address = address;
    }

```

```
    @Override
    public String toString() {
        return name + " " + address;
    }
}

//////////////////////////////////////////////

public class Student {
    private String name;
    private int age;
    private double gpa;

    public Student(String name, int age, double gpa) {
        this.name = name;
        this.age = age;
        this.gpa = gpa;
    }

    @Override
    public String toString() {
        return "Student: \n" +
            "name= " + name +
            " age= " + age +
            " gpa= " + gpa;
    }
}

//////////////////////////////////////

import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);

        Address a = new Address("HW park", "Hyderabad", "
           Telangana");
        Person p = new Person("Kowshik", a);

        Student s1 = new Student("Vinod", 26, 10.0);

        System.out.println(p);
        System.out.println(s1);
    }
}
```

Output: —

```
D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\
6_Assignment>java Main
Kowshik Address {street= HW park City= Hyderabad State= Telangana
Student:
name= Vinod age= 26 gpa= 10.0
```

Problem 10: Aggregates Library

Task: Implement a Java program that models a Library. Create classes for Library, Book, and Author. Ensure that the Library class aggregates a collection of Book objects, and each Book object has an aggregation relationship with an Author object.

Code:

```java
public class Author {
    private String name;
    private String email;

    public Author (String name, String email) {
        this.name = name;
        this.email = email;
    }

    // getters
    public String getName(){
        return name;
    }
    public  String getEmail(){
        return email;
    }

    @Override
    public String toString() {
        return name + " " + email;
    }
}

////////////////////////////////////////

public class Book {
    private String title;
    // Aggregation: Book has an Author
    private Author author;

    public Book(String title, Author author) {
        this.title = title;
        this.author = author;
    }

    // Getters
    public String getTitle() {
        return title;
```

```
    }

    public Author getAuthor() {
        return author;
    }

    @Override
    public String toString() {
        return "\'" + title + "\'" + author;
    }
}

//////////////////////////////////////

import java.util.List;
import java.util.ArrayList;

public class Library {
    private String name;
    // Aggregation: Library has a collection of Books
    private List<Book> books;

    public Library(String name) {
        this.name = name;
        books = new ArrayList<>();
    }

    public void addBook(Book book) {
        books.add(book);
    }

    public void displayBooks() {
        System.out.println("Library: " + name);
        if (books.isEmpty()) {
            System.out.println("No books in the library");
            return;
        }
        System.out.println("Books Avalible: ");
        for(Book book: books) {
            System.out.println("- " + book);
        }
    }

}

//////////////////////////////////////////

import java.util.Scanner;

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

```
90
91          Author author1 = new Author("Kowshik", "kowshik@gmail.com
                ");
92          Author author2 = new Author("Vinod", "vinod@gmail.com");
93
94          Book book1 = new Book("Python", author1);
95          Book book2 = new Book("Java", author2);
96          Book book3 = new Book("HTML", author1);
97
98          Library library = new Library("CDAC Library");
99          library.addBook(book1);
100         library.addBook(book2);
101         library.addBook(book3);
102
103         library.displayBooks();
104     }
105 }
```

Output: —

```
D:\Do_Not_Open\3_Java\Assignment_Problems\1_Assignments\1_Assignments\
6_Assignment>java Main
Library: CDAC Library
Books Avalible:
- 'Python'Kowshik kowshik@gmail.com
- 'Java'Vinod vinod@gmail.com
- 'HTML'Kowshik kowshik@gmail.com
```