OUR EDGE IS YOUR SUCCESS

Java Assignment 2 – OOP Concepts

Q1:

- Create a class **Student** with two attributes: name and age.
- Initialize them using a constructor.
- Create a method printInfo() to print the data.
- Create an object in main() and display the data.

Expected Output:
Name: Ahmed, Age: 20

Q2:

- Create a class **Animal** with a method eat() that prints "Animal is eating".
- Inherit from **Animal** to create a **Dog** class with a method bark() that prints "Dog is barking".
- In main(), create a Dog object and call both methods.

Expected Output:
Animal is eating
Dog is barking

Q3:

- Create a parent class **Vehicle** with a method display() that prints "This is a Vehicle".
- Create a child class **Car** that **overrides** the display() method to print "This is a Car".
- In main(), create a Vehicle reference pointing to a Car object and call display().

Expected Output:
This is a Car

Q4:

- Create a class **Counter** that uses a **static variable** to count how many objects are created.
- Print the total number of objects after creating three objects.

Expected Output:
Objects created: 3

Q5:

- Create a class **MathUtils** with a **static method** square(int number) that returns the square of a number.
- Call this method in main() without creating an object.

Expected Output:
Square of 5: 25

Q6:

- Create a program that declares a **final variable** PI = 3.14159.
- Print the value of PI.

(Demonstrate that changing it causes a compilation error.)

Expected Output:
Value of PI: 3.14159

Q7:

- Create a parent class **Parent** with a **final method** show() that prints "Final method in Parent".
- Create a child class and try overriding it (it should cause an error, so comment it out).
- Call the original method.

Expected Output:
Final method in Parent

Q8: Final Class

- Create a **final class** Constants with a **final variable** E = 2.718.
- Print the value of E.

(Show that extending the class is not allowed.)

Expected Output:
Value of E: 2.718

Q9:

- Create a class **Employee** with an attribute name.
- Use the this keyword in the constructor to differentiate between instance variables and parameters.
- Print employee name in a method.

Expected Output:
Employee name: Sarah

Q10:

- Create a class **Product** with two constructors:
- A default constructor that calls another constructor using this() and passes "Unknown", 0.0.
- A parameterized constructor to set name and price.
- Print product details.

Expected Output:
Product: Unknown, Price: 0.0
Product: Laptop, Price: 800.0

Q11:

- Create a parent class **Animal2** with a variable type = "General Animal".
- Create a child class **Cat** with a variable type = "Cat".
- Use super.type in a method to print both parent and child variables.

Expected Output:
Child type: Cat
Parent type: General Animal

Q12:

- Create a parent class **Vehicle2** with a constructor that prints "Vehicle brand: " + brand.
- Create a child class **Truck** that calls the parent constructor using super(brand).

Expected Output:
Vehicle brand: Volvo

Q13:

- Create a class **Calculator** with two add() methods:
- add(int a, int b) → returns the sum of integers.
- add(double a, double b) → returns the sum of doubles.
- Call both methods and print results.

Expected Output:
Sum of ints: 12
Sum of doubles: 12.8

Q14:

- Create a parent class **Shape** with a method draw() that prints "Drawing Shape".
- Create a child class **Circle** that overloads draw(int radius) to print "Drawing Circle with radius: " + radius.

Expected Output:
Drawing Shape
Drawing Circle with radius: 5

Q15: Library Management System

Requirement:

- Create a program for a **Library Management System** that demonstrates all OOP concepts:

Inheritance:

Parent class LibraryItem with attributes: title, itemId.

Child class Book adds author.

Static Variable:

Add a static variable in LibraryItem to count how many items are created.

Final Variable & Final Method:

final variable libraryName = "Central Library" final method displayLibraryInfo() printing library name (cannot be overridden).

this Keyword:

Use this to differentiate between instance variables and constructor parameters in Book.

super Keyword:

Use super() to call the parent constructor.

Method Overloading:

Overload displayInfo() in Book: one basic, one with a boolean parameter for detailed display.

Expected Output:

```
Library Name: Central Library
Title: Java Programming, ID: 101, Author: John Doe
Detailed Info:
Title: Java Programming
ID: 101
Author: John Doe
Total Library Items: 1
```

Thank You
Edges For Training Team