

# DEPARTMENT OF MCA

**Enterprise Java Lab (22MCA207)**

# Assignment - 4

17-02-2025

1. Write a Java program to reverse a given string using StringBuilder.
2. Create a program that counts and prints the number of vowels and consonants in a given string.
3. Write a program to check if two strings are anagrams of each other (contain the same characters in a different order).
4. Design an abstract class Shape with an abstract method area(). Implement two concrete classes Circle and Rectangle that compute and return their respective areas.
5. Create an abstract class Vehicle with an abstract method start(). Implement Car and Bike classes that override start() to display different starting mechanisms.
6. Implement an abstract class BankAccount with an abstract method accountType(). Derive SavingsAccount and CurrentAccount classes that provide their own implementation of accountType().
7. Implement a Person class with attributes name and age. Create a subclass Employee that adds an employeeId attribute and a method to display employee details. Demonstrate single inheritance.
8. Design a class hierarchy where Animal is the base class, Mammal is derived from Animal, and Dog is derived from Mammal. Show how multilevel inheritance works.
9. Write a program to demonstrate dynamic method dispatch where a base class Shape has a method draw(), and derived classes Circle and Rectangle override draw(). Use a parent class reference to call overridden methods.
10. Create a Payment class with a method processPayment(). Derive CreditCardPayment and PayPalPayment classes that override processPayment(). Show how dynamic method dispatch works using a parent class reference.
11. Demonstrate runtime polymorphism by creating an Animal class with a makeSound() method. Derive Dog and Cat classes that override makeSound(). Call the overridden methods using a base class reference.
12. Write a Java program that calculates interest for different types of loans. Create a base class Loan with a method getInterestRate(), and derive HomeLoan and CarLoan classes that override getInterestRate(). Show runtime polymorphism by calling overridden methods using a base class reference.
13. Create a parent class Animal with a method makeSound(). Override this method in a child class Dog and use super.makeSound() to call the parent class method.
14. Write a program where a parent class Person has an instance variable name, and a child class Employee has its own name variable. Use super.name to differentiate between them.
15. A supermarket has general employees and cashiers. Each employee has a name and an employee ID, while a cashier has an additional attribute for the cash register number. Write a Java program to demonstrate the use of the super keyword to call a parent class constructor from a child class constructor.

* Create a parent class Employee with attributes name and employeeId, and a constructor to initialize them.
* Create a child class Cashier that inherits from Employee and has an additional attribute registerNumber.
* The Cashier class should use super(name, employeeId) in its constructor to initialize the attributes of Employee.
* Implement a method in the Cashier class to display the cashier’s details.
* In the main method, create an instance of Cashier and display its details.

Your program should correctly demonstrate constructor chaining using super.

1. Create a Java program to demonstrate the use of a static variable to count the number of objects created from a class.
2. Create a Java program to demonstrate a static method that calculates the square of a number without creating an object of the class.
3. Demonstrate the use of a static block that executes before the main method in a Java program.
4. Implement a program where a static method accesses both static and non-static variables.
5. Write a Java program where a static method is inherited and accessed using both the child and parent class names.
6. Create a static nested class inside an outer class and demonstrate how it can be accessed without creating an instance of the outer class.