**13.** **class Staff {**

**int id;**

**String name, dob, joiningDate;**

**double salary;**

**Staff(int id, String name, String dob, String joiningDate, double salary) {**

**this.id = id;**

**this.name = name;**

**this.dob = dob;**

**this.joiningDate = joiningDate;**

**this.salary = salary;**

**}**

**void display() {**

**System.out.println("ID: " + id + ", Name: " + name + ", DOB: " + dob +**

**", Joining Date: " + joiningDate + ", Salary: " + salary);**

**}**

**}**

**class TeachingStaff extends Staff {**

**String[] subjects;**

**int experience;**

**TeachingStaff(int id, String name, String dob, String joiningDate, double salary,**

**String[] subjects, int experience) {**

**super(id, name, dob, joiningDate, salary);**

**this.subjects = subjects;**

**this.experience = experience;**

**}**

**void display() {**

**super.display();**

**System.out.print("Subjects: ");**

**for (String s : subjects) System.out.print(s + " ");**

**System.out.println("\nExperience: " + experience + " years");**

**}**

**}**

**class NonTeachingStaff extends Staff {**

**String department, shift;**

**NonTeachingStaff(int id, String name, String dob, String joiningDate, double salary,**

**String department, String shift) {**

**super(id, name, dob, joiningDate, salary);**

**this.department = department;**

**this.shift = shift;**

**}**

**void display() {**

**super.display();**

**System.out.println("Department: " + department + ", Shift: " + shift);**

**}**

**}**

**public class StaffTest {**

**public static void main(String[] args) {**

**Staff[] deptStaff = new Staff[4];**

**deptStaff[0] = new TeachingStaff(1, "John", "1990-01-01", "2015-06-10", 50000,**

**new String[]{"Java", "Python"}, 8);**

**deptStaff[1] = new NonTeachingStaff(2, "Alice", "1985-04-20", "2012-09-01", 30000,**

**"Admin", "Morning");**

**deptStaff[2] = new TeachingStaff(3, "David", "1992-03-15", "2017-07-15", 45000,**

**new String[]{"DBMS", "OS"}, 6);**

**deptStaff[3] = new NonTeachingStaff(4, "Emma", "1988-12-30", "2010-02-25", 28000,**

**"Library", "Evening");**

**for (Staff s : deptStaff) {**

**System.out.println("----------");**

**s.display();**

**}**

**}**

**}**

**14.** **class Employee {**

**String name, designation;**

**double salary;**

**static int count = 0;**

**// Default constructor**

**Employee() {**

**name = "Not set";**

**designation = "Not set";**

**salary = 0;**

**count++;**

**}**

**// Parameterized constructor**

**Employee(String name, String designation, double salary) {**

**this.name = name;**

**this.designation = designation;**

**this.salary = salary;**

**count++;**

**}**

**// toString override**

**public String toString() {**

**return "Name: " + name + ", Designation: " + designation + ", Salary: " + salary;**

**}**

**// Static method to get count**

**static void displayCount() {**

**System.out.println("Total Employee Objects: " + count);**

**}**

**}**

**public class EmployeeTest {**

**public static void main(String[] args) {**

**Employee e1 = new Employee("John", "Manager", 50000);**

**System.out.println(e1);**

**Employee.displayCount();**

**Employee e2 = new Employee("Alice", "Developer", 40000);**

**System.out.println(e2);**

**Employee.displayCount();**

**Employee e3 = new Employee("Bob", "Tester", 35000);**

**System.out.println(e3);**

**Employee.displayCount();**

**}**

**}**

**15.** **class InvalidAgeException extends Exception {**

**public InvalidAgeException(String msg) {**

**super(msg);**

**}**

**}**

**public class SeniorCitizenCheck {**

**public static void main(String[] args) {**

**if (args.length < 2) {**

**System.out.println("Usage: java SeniorCitizenCheck <name> <age>");**

**return;**

**}**

**String name = args[0];**

**int age = Integer.parseInt(args[1]);**

**try {**

**if (age < 60) {**

**throw new InvalidAgeException("Age must be 60 or above for senior citizen.");**

**}**

**System.out.println(name + " is a senior citizen (Age: " + age + ")");**

**} catch (InvalidAgeException e) {**

**System.out.println("Error: " + e.getMessage());**

**}**

**}**

**}**

**16.** **import java.util.Scanner;**

**class InsufficientFundException extends Exception {**

**public InsufficientFundException(String msg) {**

**super(msg);**

**}**

**}**

**public class Bank {**

**public static void main(String[] args) {**

**int balance = 1000;**

**Scanner sc = new Scanner(System.in);**

**System.out.print("Enter amount to withdraw: ");**

**int amount = sc.nextInt();**

**try {**

**if (amount > balance)**

**throw new InsufficientFundException("Insufficient funds. Available: " + balance);**

**balance -= amount;**

**System.out.println("Withdrawn: " + amount);**

**System.out.println("Remaining Balance: " + balance);**

**} catch (InsufficientFundException e) {**

**System.out.println("Error: " + e.getMessage());**

**}**

**}**

**}**

**17.class PrimeThread extends Thread {**

**public void run() {**

**for (int i = 2; i <= 500; i++) {**

**if (isPrime(i)) {**

**System.out.println("Prime: " + i);**

**try { Thread.sleep(3000); } catch (Exception e) {}**

**}**

**}**

**}**

**boolean isPrime(int n) {**

**for (int i = 2; i <= Math.sqrt(n); i++)**

**if (n % i == 0) return false;**

**return true;**

**}**

**public static void main(String[] args) {**

**new PrimeThread().start();**

**}**

**}**

**18.**class PerfectNumberThread extends Thread {

public void run() {

for (int num = 1; num <= 1000; num++) {

int sum = 0;

for (int i = 1; i < num; i++) {

if (num % i == 0) sum += i;

}

if (sum == num) {

System.out.println("Perfect Number: " + num);

try {

Thread.sleep(5000); // wait for 5 seconds

} catch (InterruptedException e) {

System.out.println("Interrupted!");

}

}

}

}

public static void main(String[] args) {

PerfectNumberThread t = new PerfectNumberThread();

t.start();

}

}

**19.** class NumberPrinter {

public synchronized void printSmallNumbers() {

for (int i = 5; i <= 25; i += 5) {

System.out.println(i);

try { Thread.sleep(100); } catch (InterruptedException e) {}

}

}

public synchronized void printLargeNumbers() {

for (int i = 100; i <= 500; i += 100) {

System.out.println(i);

try { Thread.sleep(100); } catch (InterruptedException e) {}

}

}

}

public class SyncExample {

public static void main(String[] args) {

NumberPrinter printer = new NumberPrinter();

Thread t1 = new Thread(() -> printer.printSmallNumbers());

Thread t2 = new Thread(() -> printer.printLargeNumbers());

t1.start();

try { t1.join(); } catch (InterruptedException e) {} // wait for t1 to finish

t2.start();

}

}