----------------problem 1------------------

import java.util.Scanner;

public class SquareRootCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

// Taking user input

System.out.print("Enter a number: ");

double number = scanner.nextDouble();

// Checking if the number is negative

if (number < 0) {

throw new IllegalArgumentException("Error: Cannot calculate the square root of a negative number.");

}

// Calculating and displaying square root

double result = Math.sqrt(number);

System.out.println("Square root: " + result);

} catch (IllegalArgumentException e) {

System.out.println(e.getMessage());

} catch (Exception e) {

System.out.println("Error: Invalid input. Please enter a numeric value.");

} finally {

scanner.close(); // Closing scanner to prevent resource leak

}

}

}

------------------problem 2-----------------

import java.util.Scanner;

class InsufficientBalanceException extends Exception {

public InsufficientBalanceException(String message) {

super(message);

}

}

class InvalidPinException extends Exception {

public InvalidPinException(String message) {

super(message);

}

}

public class ATMSimulator {

private static final String CORRECT\_PIN = "1234";

private static double balance = 3000.00;

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.print("Enter PIN: ");

String enteredPin = scanner.nextLine();

validatePin(enteredPin);

System.out.print("Withdraw Amount: ");

double withdrawalAmount = scanner.nextDouble();

withdraw(withdrawalAmount);

} catch (InvalidPinException e) {

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

} catch (InsufficientBalanceException e) {

System.out.println("Error: " + e.getMessage() + " Current Balance: " + balance);

} finally {

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

scanner.close();

}

}

private static void validatePin(String pin) throws InvalidPinException {

if (!pin.equals(CORRECT\_PIN)) {

throw new InvalidPinException("Invalid PIN.");

}

}

private static void withdraw(double amount) throws InsufficientBalanceException {

if (amount > balance) {

throw new InsufficientBalanceException("Insufficient balance.");

}

balance -= amount;

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

}

}

------------------problem 3---------------------

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class CourseFullException extends Exception {

public CourseFullException(String message) {

super(message);

}

}

class PrerequisiteNotMetException extends Exception {

public PrerequisiteNotMetException(String message) {

super(message);

}

}

class Course {

private String name;

private int maxEnrollment;

private List<String> prerequisites;

private List<String> enrolledStudents;

public Course(String name, int maxEnrollment, List<String> prerequisites) {

this.name = name;

this.maxEnrollment = maxEnrollment;

this.prerequisites = prerequisites;

this.enrolledStudents = new ArrayList<>();

}

public String getName() {

return name;

}

public boolean isFull() {

return enrolledStudents.size() >= maxEnrollment;

}

public void enroll(String student, List<String> completedCourses) throws CourseFullException, PrerequisiteNotMetException {

if (isFull()) {

throw new CourseFullException("Course is full. Cannot enroll in " + name + ".");

}

for (String prerequisite : prerequisites) {

if (!completedCourses.contains(prerequisite)) {

throw new PrerequisiteNotMetException("Complete " + prerequisite + " before enrolling in " + name + ".");

}

}

enrolledStudents.add(student);

System.out.println("Enrollment successful in " + name + ".");

}

}

public class UniversityEnrollmentSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Creating a course with prerequisites

List<String> prerequisites = new ArrayList<>();

prerequisites.add("Core Java");

Course advancedJava = new Course("Advanced Java", 2, prerequisites);

System.out.print("Enter your name: ");

String studentName = scanner.nextLine();

System.out.print("Enter completed courses (comma separated): ");

String completedCoursesInput = scanner.nextLine();

List<String> completedCourses = List.of(completedCoursesInput.split(", "));

try {

advancedJava.enroll(studentName, completedCourses);

} catch (CourseFullException | PrerequisiteNotMetException e) {

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

} finally {

System.out.println("Enrollment process completed.");

scanner.close();

}

}

}