Assignment 1: Java Fundamentals

```
import java.util.Scanner;
public class BasicArithmetic {

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

        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();

        System.out.print("Enter second number: ");
        double num2 = scanner.nextDouble();

        System.out.println("\nSelect operation:");
        System.out.println("1. Addition");
        System.out.println("2. Subtraction");
        System.out.println("3. Multiplication");
        System.out.println("4. Division");

        System.out.print("Enter operation (1/2/3/4): ");
        int operation = scanner.nextInt();

        double result;

        switch (operation) {
            case 1:
                result = num1 + num2;
                System.out.println("The result of " + num1 + " + " + num2 + " = " + result);
                break;
            case 2:
                result = num1 - num2;
```

```
            System.out.println("The result of " + num1 + " - " + num2 + " = " + result);
            break;
        case 3:
            result = num1 * num2;
            System.out.println("The result of " + num1 + " * " + num2 + " = " + result);
            break;
        case 4:
            if (num2 == 0) {
                System.out.println("Error! Division by zero.");
            } else {
                result = num1 / num2;
                System.out.println("The result of " + num1 + " / " + num2 + " = " + result);
            }
            break;
        default:
            System.out.println("Invalid operation selected.");
    }

    scanner.close();
  }
}
```

```
import java.util.Scanner;

public class EvenOddChecker {

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

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

```
        int num = scanner.nextInt();

        if (num % 2 == 0) {
            System.out.println(num + " is an even number.");
        } else {
            System.out.println(num + " is an odd number.");
        }

        scanner.close();
    }
}
```

```
import java.util.Scanner;

public class SimpleCalculator {

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

        System.out.println("Simple Calculator");
        System.out.println("Select operation:");
        System.out.println("1. Addition");
        System.out.println("2. Subtraction");
        System.out.println("3. Multiplication");
        System.out.println("4. Division");
        System.out.print("Enter operation (1/2/3/4): ");
        int operation = scanner.nextInt();
        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();
       System.out.print("Enter second number: ");
```

```
double num2 = scanner.nextDouble();

double result;

switch (operation) {
    case 1:
        result = num1 + num2;
        System.out.println("The result of " + num1 + " + " + num2 + " = " + result);
        break;
    case 2:
        result = num1 - num2;
        System.out.println("The result of " + num1 + " - " + num2 + " = " + result);
        break;
    case 3:
        result = num1 * num2;
        System.out.println("The result of " + num1 + " * " + num2 + " = " + result);
        break;
    case 4:
        if (num2 == 0) {
            System.out.println("Error! Division by zero.");
        } else {
            result = num1 / num2;
            System.out.println("The result of " + num1 + " / " + num2 + " = " + result);
        }
        break;
    default:
        System.out.println("Invalid operation selected.");
}

scanner.close();
```

Assignment 2: Classes, Objects, and Methods

```
class Person {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void displayAttributes() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

public class Main {
    public static void main(String[] args) {
        Person person1 = new Person("Alice", 30);
        Person person2 = new Person("Bob", 25);

        person1.displayAttributes();
        person2.displayAttributes();
    }
}
```

```
import java.util.Scanner;

class BankAccount {
    private double balance;
```

```
    public BankAccount() {
        balance = 0.0;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else if (amount > balance) {
            System.out.println("Insufficient funds for withdrawal.");
        } else {
            System.out.println("Withdrawal amount must be positive.");
        }
    }

    public void displayBalance() {
        System.out.println("Current balance: $" + balance);
    }
}

public class Main {
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    BankAccount account = new BankAccount();
    int choice;

    do {
        System.out.println("\nBank Account Menu:");
        System.out.println("1. Deposit");
        System.out.println("2. Withdraw");
        System.out.println("3. Display Balance");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter amount to deposit: ");
                double depositAmount = scanner.nextDouble();
                account.deposit(depositAmount);
                break;
            case 2:
                System.out.print("Enter amount to withdraw: ");
                double withdrawAmount = scanner.nextDouble();
                account.withdraw(withdrawAmount);
                break;
            case 3:
                account.displayBalance();
                break;
            case 4:
                System.out.println("Exiting...");
                break;
```

```
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 4);

    scanner.close();
  }
}
```

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

  // Constructor with two parameters
  public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
  }

  // Constructor with one parameter (for square)
  public Rectangle(double side) {
    this.length = side;
    this.width = side;
  }

  // Method to calculate area with length and width
  public double area() {
    return length * width;
  }
```

```
    // Overloaded method to calculate area using a single parameter
    public double area(double side) {
        return side * side;
    }

    public void display() {
        System.out.println("Length: " + length + ", Width: " + width);
        System.out.println("Area: " + area());
    }
}

public class Main {
    public static void main(String[] args) {
        Rectangle rectangle1 = new Rectangle(5.0, 3.0);
        Rectangle rectangle2 = new Rectangle(4.0); // Square

        rectangle1.display();
        rectangle2.display();

        // Using the overloaded area method
        double areaOfSquare = rectangle2.area(4.0);
        System.out.println("Area of the square with side 4.0: " + areaOfSquare);
    }
}
```

Assignment 3: Advanced OOP Concepts

```
class Vehicle {
public void start() {
        System.out.println("Vehicle is starting.");
    }

    public void stop() {
        System.out.println("Vehicle is stopping.");
    }
}

class Car extends Vehicle {
    @Override
    public void start() {
        System.out.println("Car is starting with a key.");
    }

    @Override
    public void stop() {
        System.out.println("Car is stopping with brakes.");
    }
}

class Bike extends Vehicle {
    @Override
    public void start() {
        System.out.println("Bike is starting with a push button.");
    }

    @Override
```

```
    public void stop() {
        System.out.println("Bike is stopping with rear brakes.");
    }
}

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

        myVehicle = new Car();
        myVehicle.start();
        myVehicle.stop();

        myVehicle = new Bike();
        myVehicle.start();
        myVehicle.stop();
    }
}
```

```
import java.util.Scanner;

abstract class Shape {
    abstract double area();
}

class Circle extends Shape {
    private double radius;

    public Circle(double radius) {
```

```
        this.radius = radius;
    }

    @Override
    double area() {
        return Math.PI * radius * radius;
    }
}

class Rectangle extends Shape {
    private double length;
    private double width;

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

    @Override
    double area() {
        return length * width;
    }
}

class Triangle extends Shape {
    private double base;
    private double height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
```

```
    }

    @Override
    double area() {
        return 0.5 * base * height;
    }
}

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

        System.out.print("Enter radius of the circle: ");
        double radius = scanner.nextDouble();
        Shape circle = new Circle(radius);
        System.out.println("Area of the circle: " + circle.area());

        System.out.print("Enter length and width of the rectangle: ");
        double length = scanner.nextDouble();
        double width = scanner.nextDouble();
        Shape rectangle = new Rectangle(length, width);
        System.out.println("Area of the rectangle: " + rectangle.area());

        System.out.print("Enter base and height of the triangle: ");
        double base = scanner.nextDouble();
        double height = scanner.nextDouble();
        Shape triangle = new Triangle(base, height);
        System.out.println("Area of the triangle: " + triangle.area());
        scanner.close();
    }
}
```

```
abstract class Animal {
    abstract void sound();
}

// Interface
interface Playable {
    void play();
}

class Dog extends Animal implements Playable {
    @Override
    void sound() {
        System.out.println("Dog barks: Woof!");
    }

    @Override
    public void play() {
        System.out.println("Dog plays fetch.");
    }
}

class Cat extends Animal implements Playable {
    @Override
    void sound() {
        System.out.println("Cat meows: Meow!");
    }

    @Override
    public void play() {
```

```
        System.out.println("Cat chases a laser pointer.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal dog = new Dog();
        Animal cat = new Cat();

        dog.sound();
        cat.sound();

        Playable playableDog = (Playable) dog;
        Playable playableCat = (Playable) cat;

        playableDog.play();
        playableCat.play();
    }
}
```

Assignment 4: Working with Java Collections and Handling Exceptions

```
import java.util.ArrayList;
import java.util.Scanner;

class Student {
    private String name;
    private int id;

    public Student(String name, int id) {
        this.name = name;
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public int getId() {
        return id;
    }

    @Override
    public String toString() {
        return "Student ID: " + id + ", Name: " + name;
    }
}

public class StudentManager {
    private ArrayList<Student> students;
```

```
public StudentManager() {
    students = new ArrayList<>();
}

public void addStudent(String name, int id) {
    Student newStudent = new Student(name, id);
    students.add(newStudent);
    System.out.println("Student added: " + newStudent);
}

public void removeStudent(int id) {
    for (Student student : students) {
        if (student.getId() == id) {
            students.remove(student);
            System.out.println("Student removed: " + student);
            return;
        }
    }
    System.out.println("Student with ID " + id + " not found.");
}

public void displayStudents() {
    if (students.isEmpty()) {
        System.out.println("No students in the list.");
    } else {
        System.out.println("List of Students:");
        for (Student student : students) {
            System.out.println(student);
        }
    }
```

```
}

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

    do {
        System.out.println("\nStudent Management Menu:");
        System.out.println("1. Add Student");
        System.out.println("2. Remove Student");
        System.out.println("3. Display Students");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine();  // Consume newline

        switch (choice) {
            case 1:
                System.out.print("Enter student name: ");
                String name = scanner.nextLine();
                System.out.print("Enter student ID: ");
                int id = scanner.nextInt();
                manager.addStudent(name, id);
                break;
            case 2:
                System.out.print("Enter student ID to remove: ");
                int removeId = scanner.nextInt();
                manager.removeStudent(removeId);
                break;
            case 3:
```

```
                manager.displayStudents();
                break;
            case 4:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 4);

    scanner.close();
  }
}
```

```
import java.util.HashMap;
import java.util.Scanner;

class Employee {
  private String name;
  private String department;

  public Employee(String name, String department) {
    this.name = name;
    this.department = department;
  }

  @Override
  public String toString() {
    return "Name: " + name + ", Department: " + department;
  }
```

```
}

public class EmployeeManager {
    private HashMap<Integer, Employee> employeeMap;

    public EmployeeManager() {
        employeeMap = new HashMap<>();
    }

    public void addEmployee(int id, String name, String department) {
        Employee employee = new Employee(name, department);
        employeeMap.put(id, employee);
        System.out.println("Employee added: ID = " + id + ", " + employee);
    }

    public void getEmployee(int id) {
        Employee employee = employeeMap.get(id);
        if (employee != null) {
            System.out.println("Employee details: ID = " + id + ", " + employee);
        } else {
            System.out.println("Employee with ID " + id + " not found.");
        }
    }

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

        do {
            System.out.println("\nEmployee Management Menu:");
```

```
        System.out.println("1. Add Employee");
        System.out.println("2. Get Employee Details");
        System.out.println("3. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine();  // Consume newline

        switch (choice) {
            case 1:
                System.out.print("Enter employee ID: ");
                int id = scanner.nextInt();
                scanner.nextLine();  // Consume newline
                System.out.print("Enter employee name: ");
                String name = scanner.nextLine();
                System.out.print("Enter employee department: ");
                String department = scanner.nextLine();
                manager.addEmployee(id, name, department);
                break;
            case 2:
                System.out.print("Enter employee ID to retrieve details: ");
                int searchId = scanner.nextInt();
                manager.getEmployee(searchId);
                break;
            case 3:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 3);
```

```
        scanner.close();
    }
}
```

```
import java.util.Scanner;
// Custom Exception
class InvalidInputException extends Exception {
    public InvalidInputException(String message) {
        super(message);
    }
}

public class ExceptionHandlingExample {

    public static double divide(int a, int b) throws ArithmeticException {
        return a / b;
    }

    public static void validateInput(String input) throws InvalidInputException {
        if (input == null || input.trim().isEmpty()) {
            throw new InvalidInputException("Input cannot be null or empty.");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int numerator, denominator;
        String input;

        try {
```

```
            System.out.print("Enter numerator: ");
            input = scanner.nextLine();
            validateInput(input);
            numerator = Integer.parseInt(input);

            System.out.print("Enter denominator: ");
            input = scanner.nextLine();
            validateInput(input);
            denominator = Integer.parseInt(input);

            double result = divide(numerator, denominator);
            System.out.println("Result: " + result);

        } catch (ArithmeticException e) {
            System.out.println("Error: Cannot divide by zero.");
        } catch (NumberFormatException e) {
            System.out.println("Error: Invalid number format.");
        } catch (InvalidInputException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (NullPointerException e) {
            System.out.println("Error: A null value was encountered.");
        } catch (Exception e) {
            System.out.println("An unexpected error occurred: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```
