**Assignment 5: Method Overloading and Constructors**

**Practical Problems**

1. **Temperature Converter:** Create a Java class with overloaded methods to convert

temperature from Celsius to Fahrenheit and vice versa. Include constructors

to initialize the temperature values.

2. **Calculator:** Design a Java class for a simple calculator that can perform addition,

subtraction, multiplication, and division. Implement overloaded methods

for each operation, and use constructors to initialize operands.

3. **Bank Account:** Develop a Java class for a bank account that supports deposit

and withdrawal operations. Use method overloading to handle different types of

deposits (cash, check) and constructors to initialize account details.

4. **Geometry Calculator:** Write a Java program with overloaded methods to calculate

the area of various geometric shapes such as circle, rectangle, and triangle.

Include constructors to set the dimensions of each shape.

5. **String Manipulation:** Design a Java class with overloaded methods for string

manipulation tasks such as concatenation, substring extraction, and string reversal.

Use constructors to initialize input strings.

6. **Employee Management System:** Create a Java class for managing employee

details, including name, salary, and department. Implement overloaded methods

for adding employees with different sets of parameters, and use constructors for

initialization.

7. **Student Grading System:** Develop a Java program to handle student grades,

including methods to calculate the average, highest, and lowest grades. Use

method overloading to support different input formats (array, list) and constructors

to initialize student data.

8. **Shape Drawer:** Write a Java class for drawing shapes on a graphical interface,

such as circles, rectangles, and triangles. Implement overloaded methods for

drawing shapes with different parameters, and use constructors to set properties.

9. **File Operations:** Design a Java class for performing file operations like reading,

writing, and copying files. Use method overloading to handle different file types

(text, binary) and constructors to initialize file paths.

10. **Time Management:** Develop a Java program to manage time-related tasks,

including methods for adding, subtracting, and comparing time intervals. Use

method overloading to support different time formats (hours, minutes, seconds)

and constructors to set initial time values.