Learner Assignment Submission Format

Learner Details

- **Name: Akash**
- **Enrollment Number:**
- **Batch / Class:**
- **Assignment: (Bridge Course Day 1)**
- **Date of Submission:**

Problem Solving Activity 1.1

1.We have to take take name and the age as input and create a greeting like "hello (name),you are (age) years old"

---

2. Algorithm

1. Initialize variables:

   - name = "akash"

   - age = 21

2. Print greeting message:

   - Combine "hello ", name, " you are ", age, and " years old" into a single string

   - Output the resulting string

---

3. Pseudocode

```
BEGIN
  SET name = "akash"
  SET age = 21
  PRINT "hello " + name + " you are " + age + " years old"
```

END

4. Program Code

```
class D1_1{

    public static void main(String[]args){

       String name="akash";

       int age=21;

        System.out.println("hello "+name+" you are "+age+" years old " );



    }



}
```

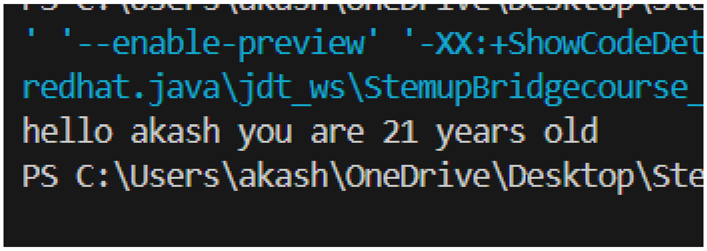
5. Test Cases

Present a table of test cases you used to validate your program. Include a mix of regular, boundary, and edge cases.

Test Case No.	Input	Expected Output	Actual Output	Status (Pass/Fail)
1	Akash 21	hello akash you are 21 years old	hello akash you are 21 years old	pass
2	Akash 31	hello akash you are 31 years old	hello akash you are 31 years old	pass

3	Akash 45	hello akash you are 45 years old	hello akash you are 45 years old	pass

6. Screenshots of Output



```
PS C:\Users\akash\OneDrive\Desktop\Ste
' '--enable-preview' '-XX:+ShowCodeDet
redhat.java\jdt_ws\StemupBridgecourse_
hello akash you are 21 years old
PS C:\Users\akash\OneDrive\Desktop\Ste
```

7. Observation / Reflection

This Java program demonstrates basic variable declaration, assignment, and string concatenation. The code is straightforward and effectively prints out a personalized greeting message

Problem Solving Activity 1.2

1.Take two numbers as input (strings), convert them to integers, and print their sum, difference, and product.

2. Algorithm

. Take two numbers as input from the user as strings.

. Convert the input strings to integers.

. Calculate the sum, difference, and product of the two numbers.

. Print the results.

3. Pseudocode

```
BEGIN
  SET input1 = "num1"
  SET input2 = "num2"
  CONVERT input1 TO INTEGER num1
  CONVERT input2 TO INTEGER num2
  CALCULATE sum = num1 + num2
  PRINT sum
END
```

4. Program Code

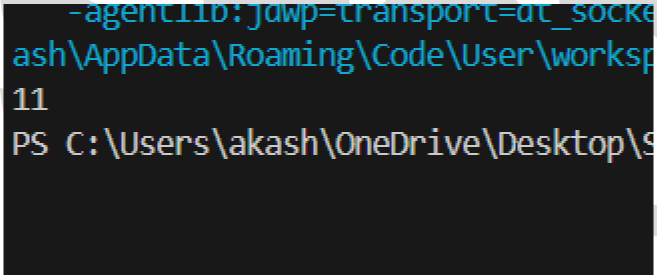
```
class D1_2{
    public static void main(String[]args){
        String input1="5";
        String input2="6";

System.out.println(Integer.parseInt(input1)+Integer.parseInt(input2) );



    }
}
```

5. Test Cases

Present a table of test cases you used to validate your program. Include a mix of regular, boundary, and edge cases.

Test Case No.	Input	Expected Output	Actual Output	Status (Pass/Fail)
1	5,6	11	11	pass
2	4,8	12	12	pass
3	34,56	90	90	pass

6. Screenshots of Output



```
   -agentlib:jdwp=transport=dt_socke
ash\AppData\Roaming\Code\User\worksp
11
PS C:\Users\akash\OneDrive\Desktop\S
```

7. Observation / Reflection

This Java code demonstrates how to convert string representations of numbers to integers and perform arithmetic operations on them. The Integer.parseInt() method is used to convert the string inputs to integers, which can then be added together.

Problem Solving Activity 1.3

1.Identify the data type of the following inputs in your language choice: "123", 123, 123.45, True, "Hello"

2. Algorithm

. Check if the input is a string (enclosed in quotes):

- If yes, the data type is String.

. Check if the input is a whole number (without decimal points):

- If yes, the data type is int (or long if it's a large number).

. Check if the input is a decimal number:

- If yes, the data type is double (or float if it's explicitly declared).

. Check if the input is a boolean value (true or false):

- If yes, the data type is boolean.

3. Pseudocode

```
IF input IS STRING
  THEN data type = String
ELSE IF input IS WHOLE NUMBER
  THEN data type = int (or long)
ELSE IF input IS DECIMAL NUMBER
  THEN data type = double (or float)
ELSE IF input IS BOOLEAN VALUE
  THEN data type = boolean
ELSE
  THEN data type = unknown
```

4. Program Code

"123": String

- 123: int

- 123.45: double

- True: boolean

- "Hello": String

7. Observation / Reflection

Understanding data types is crucial in programming, as it determines how data is stored, processed, and manipulated. In Java, each data type has its own specific characteristics and uses.

Problem Solving Activity 1.4

1. a program that converts Celsius to Fahrenheit using a variable and formula:

F= (C*9/5)+32.

2. Algorithm

1. Start

2. Declare variables: C (Celsius), F (Fahrenheit)

3. Assign value to C

4. Calculate F = (C * 9 / 5) + 32

5. Print F

6. End

3. Pseudocode

```
BEGIN
  SET celsius = temperature value
  CALCULATE fahrenheit = (celsius * 9 / 5) + 32
  PRINT "Celsius temperature: " + celsius
  PRINT "Fahrenheit temperature: " + fahrenheit
END
```

4. Program Code

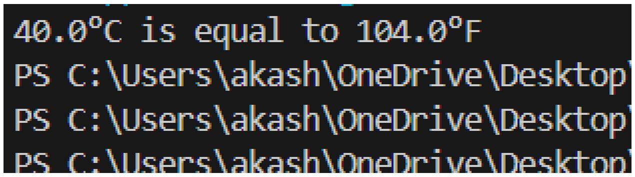
```
public class D1_4 {


    public static void main(String[] args) {

        double celsius = 30.0;

        double fahrenheit = (celsius * 9 / 5) + 32;
        System.out.println(celsius + "°C is equal to " + fahrenheit +
"°F");
    }
}
```

5. Test Cases

Present a table of test cases you used to validate your program. Include a mix of regular, boundary, and edge cases.

Test Case No.	Input	Expected Output	Actual Output	Status (Pass/Fail)
1	30	86	86	pass
2	40	104	104	pass
3	26	78.8	78.8	pass

6. Screenshots of Output



```
40.0°C is equal to 104.0°F
PS C:\Users\akash\OneDrive\Desktop
PS C:\Users\akash\OneDrive\Desktop
PS C:\Users\akash\OneDrive\Desktop
```

7. Observation / Reflection

This temperature conversion program demonstrates a simple yet effective way to convert between Celsius and Fahrenheit scales. The formula used is a standard conversion formula that can be applied to any temperature value.

Problem Solving Activity 1.5

1.Create a basic calculator that performs ..", and/between two user provided numbers.

2. Algorithm

1. Take two numbers as input from the user.

2. Display available operations (+, -, *, /).

3. Take the operator as input from the user.

4. Perform the operation based on the operator:

   - If +, calculate the sum.

   - If -, calculate the difference.

   - If *, calculate the product.

   - If /, calculate the quotient (check for division by zero).

5. Display the result.

3. Pseudocode

```
BEGIN
  INPUT num1
  INPUT num2
  DISPLAY available operations
  INPUT operator
  SWITCH operator
    CASE '+':
      CALCULATE result = num1 + num2
      PRINT "Sum: " + result
    CASE '-':
      CALCULATE result = num1 - num2
      PRINT "Difference: " + result
    CASE '*':
      CALCULATE result = num1 * num2
      PRINT "Product: " + result
    CASE '/':
      IF num2 != 0
        CALCULATE result = num1 / num2
        PRINT "Quotient: " + result
  ELSE
        PRINT "Error: Division by zero is not allowed."
    DEFAULT:
      PRINT "Invalid operator."
END
```

4. Program Code

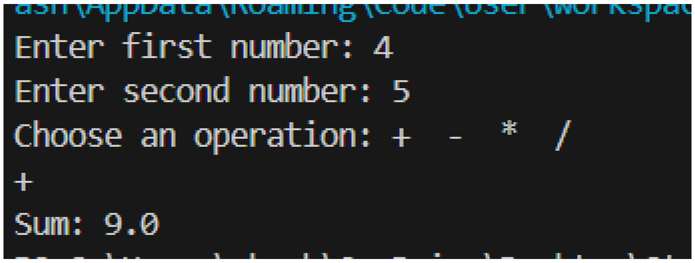
```java
import java.util.Scanner;


public class D1_5 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);



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

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


        System.out.println("Choose an operation: +  -  *  /");
        char operator = input.next().charAt(0);

        double result;

        switch (operator) {
            case '+':
                result = num1 + num2;
                System.out.println("Sum: " + result);
                break;
            case '-':
                result = num1 - num2;
                System.out.println("Difference: " + result);
                break;
            case '*':
                result = num1 * num2;
                System.out.println("Product: " + result);
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                    System.out.println("Quotient: " + result);
                } else {
                    System.out.println("Error: Division by zero is not allowed.");
                }
                break;
            default:
                System.out.println("Invalid operator.");
        }

        input.close();
    }
}
```

5. Test Cases

Present a table of test cases you used to validate your program. Include a mix of regular, boundary, and edge cases.

Test Case No.	Input	Expected Output	Actual Output	Status (Pass/Fail)
1	4,5,+	9	9	pass
2	10,5,*	50	50	pass
3	30,5,/	6	6	pass

6. Screenshots of Output



```
ash\AppData\Roaming\Code\User\workspac
Enter first number: 4
Enter second number: 5
Choose an operation: +  -  *  /
+
Sum: 9.0
```

7. Observation / Reflection

This basic calculator program demonstrates how to take user input, perform arithmetic operations, and handle potential errors (division by zero). The program uses a switch statement to determine which operation to perform based on the user's input.