

Ai Assisted coding  
LAB TEST 4  
SET-A

Question-1:

PROMPT: 1 Write a python code to calculate the factorial using recursion.  
2 convert the above code into java

Python code:

```
def factorial(n):
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers.")

    if n == 0 or n == 1:
        return 1

    return n * factorial(n - 1)

if __name__ == "__main__":
    print("--- Recursive Factorial Calculator ---")

    try:
        user_input = input("Enter a non-negative integer: ")
        num = int(user_input)

        result = factorial(num)

        print(f"The factorial of {num}! is: {result}")

    except ValueError as e:
        print(f"Error: {e}")
    except Exception as e:
        print(f"An unexpected error occurred: {e}")
```

Output:

```
--- Recursive Factorial Calculator ---
Enter a non-negative integer: 3
The factorial of 3! is: 6
```

Java code:

```
1- public class Factorial {  
2-     public static int factorial(int n) {  
3-         if (n == 0)  
4-             return 1;  
5-         else  
6-             return n * factorial(n - 1);  
7-     }  
8-  
9-     public static void main(String[] args) {  
10-         System.out.println("Factorial of 3: " + factorial(3));  
11-     }  
12- }
```

Output:

```
Factorial of 3: 6
```

Explanation:Code Structure

Python: Uses def keyword and indentation for structure

Java: Requires class encapsulation and explicit {}

bracesEntry Points

Python: Uses if `_name_ == "_main_"`:

Java: Requires public static void main(String[] args)

Type System

Python: Dynamic typing with no explicit type declarations

Java: Static typing requiring explicit types (int, long, etc.)

Integer Handling

Python: Automatic arbitrary-precision integers

Java: Limited integer types requiring explicit long or BigInteger for large factorials

Question-2:

PROMPT: 1 Task: Write a C++ program to sort an array of integers using the Bubble Sort algorithm.

2 convert the above code into python

C++ code:

```
1  #include <iostream>
2  using namespace std;
3
4  void bubbleSort(int arr[], int n) {
5      for (int i = 0; i < n-1; i++)
6          for (int j = 0; j < n-i-1; j++)
7              if (arr[j] > arr[j+1])
8                  swap(arr[j], arr[j+1]);
9  }
10
11 int main() {
12     int arr[] = {5, 2, 9, 1, 5, 6};
13     int n = sizeof(arr)/sizeof(arr[0]);
14     bubbleSort(arr, n);
15     for (int i = 0; i < n; i++)
16         cout << arr[i] << " ";
17     return 0;
18 }
```

Output:

Output
1 2 5 5 6 9

Python code :

```
4b.2.py > ...
1  def bubble_sort(arr):
2      n = len(arr)
3      for i in range(n-1):
4          for j in range(n-i-1):
5              if arr[j] > arr[j+1]:
6                  arr[j], arr[j+1] = arr[j+1], arr[j]
7
8  arr = [5, 2, 9, 1, 5, 6]
9  bubble_sort(arr)
10 print(arr)
```

Output:

```
[1, 2, 5, 5, 6, 9]
```

Explanation:

- **C++:** Fast, efficient, strong typing, harder to learn.
- **Python:** Slower, concise, easy to read and learn.
- **Best Use:**
  - C++ → Systems, games, embedded apps.
  - Python → Education, scripting, data science.