

NAME:M.AKASH ROLL NO:2303A51820 BATCH:26

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING																		
Program Name: B. Tech		Assignment Type: Lab	Academic Year: 2025-2026																	
Course Coordinator Name		Dr. Rishabh Mittal																		
Instructor(s) Name		<table border="1"> <tr><td>Mr. S Naresh Kumar</td></tr> <tr><td>Ms. B. Swathi</td></tr> <tr><td>Dr. Sasanko Shekhar Gantayat</td></tr> <tr><td>Mr. Md Sallauddin</td></tr> <tr><td>Dr. Mathivanan</td></tr> <tr><td>Mr. Y Srikanth</td></tr> <tr><td>Ms. N Shilpa</td></tr> <tr><td>Dr. Rishabh Mittal (Coordinator)</td></tr> <tr><td>Dr. R. Prashant Kumar</td></tr> <tr><td>Mr. Ankushavali MD</td></tr> <tr><td>Mr. B Viswanath</td></tr> <tr><td>Ms. Sujitha Reddy</td></tr> <tr><td>Ms. A. Anitha</td></tr> <tr><td>Ms. M.Madhuri</td></tr> <tr><td>Ms. Katherashala Swetha</td></tr> <tr><td>Ms. Velpula sumalatha</td></tr> <tr><td>Mr. Bingi Raju</td></tr> </table>		Mr. S Naresh Kumar	Ms. B. Swathi	Dr. Sasanko Shekhar Gantayat	Mr. Md Sallauddin	Dr. Mathivanan	Mr. Y Srikanth	Ms. N Shilpa	Dr. Rishabh Mittal (Coordinator)	Dr. R. Prashant Kumar	Mr. Ankushavali MD	Mr. B Viswanath	Ms. Sujitha Reddy	Ms. A. Anitha	Ms. M.Madhuri	Ms. Katherashala Swetha	Ms. Velpula sumalatha	Mr. Bingi Raju
Mr. S Naresh Kumar																				
Ms. B. Swathi																				
Dr. Sasanko Shekhar Gantayat																				
Mr. Md Sallauddin																				
Dr. Mathivanan																				
Mr. Y Srikanth																				
Ms. N Shilpa																				
Dr. Rishabh Mittal (Coordinator)																				
Dr. R. Prashant Kumar																				
Mr. Ankushavali MD																				
Mr. B Viswanath																				
Ms. Sujitha Reddy																				
Ms. A. Anitha																				
Ms. M.Madhuri																				
Ms. Katherashala Swetha																				
Ms. Velpula sumalatha																				
Mr. Bingi Raju																				
CourseCode	23CS002PC304	Course Title	AI Assisted Coding																	
Year/Sem	III/II	Regulation	R23																	
Date and Day of Assignment	Week2	Time(s)	23CSBTB01 To 23CSBTB52																	
Duration	2 Hours	Applicable to Batches	All batches																	
Assignment Number: 3.4 (Present assignment number)/ 24 (Total number of assignments)																				
Q.No.	Question	Expected Time to complete																		
1	Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques	Week2																		

Task 1: Zero-shot Prompt – Fibonacci Series Generator

Task Description #1

- Without giving an example, write a single comment prompt asking GitHub Copilot to generate a Python function to print the first N Fibonacci numbers.

Expected Output #1



```
J ai.java > Language Support for Java(TM) by Red Hat >  ai
1  import java.util.Scanner;
2
3  public class ai {
4      Run main | Debug main | Run | Debug
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7          System.out.print(s: "Enter n: ");
8          int n = scanner.nextInt();
9
10         System.out.println("First " + n + " Fibonacci numbers:");
11
12         int a = 0, b = 1;
13         for (int i = 0; i < n; i++) {
14             System.out.print(a + " ");
15             int temp = a + b;
16             a = b;
17             b = temp;
18         }
19         scanner.close();
20     }
21 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
orkspaceStorage\272c303df60776de7abd45f82063c306\redhat.java\jdt_ws\ai
Enter n: 4
First 4 Fibonacci numbers:
0 1 1 2
PS D:\ai assist> 
```

Task 2: One-shot Prompt – List Reversal Function

Task Description #2

- Write a comment prompt to reverse a list and provide one example below the comment to guide Copilot.

Expected Output #2

The screenshot shows an IDE with a Java file named `ai2.java`. The code defines a class `ai2` with a `main` method. The `main` method creates an `ArrayList` of integers, adds the elements 1, 2, 3, and 4, prints the original list, reverses the list using `Collections.reverse()`, and prints the reversed list. The terminal output shows the execution of the program, displaying the original list `[1, 2, 3, 4, 5]` and the reversed list `[5, 4, 3, 2, 1]`.

```

Welcome | J ai.java 1 | J ai2.java X
J ai2.java > Language Support for Java(TM) by Red Hat > ai2
1  /* Write a comment prompt to reverse a list and provide one
2
3  // Reverse a list in Java
4  import java.util.ArrayList;
5  import java.util.Collections;
6  import java.util.List;
7
8  public class ai2 {
    Run main | Debug main | Run | Debug
9      public static void main(String[] args) {
10         List<Integer> numbers = new ArrayList<>();
11         Collections.addAll(numbers, ...elements: 1, 2, 3, 4,
12
13         System.out.println("Original list: " + numbers);
14
15         Collections.reverse(numbers);
16
17         System.out.println("Reversed list: " + numbers);
18     }
19 }

PROBLEMS 1 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS
PS D:\ai assist> & 'C:\Program Files\Java\jdk-21.0.10\bin\java.exe' '
onMessages' '-cp' 'C:\Users\WINDOWS\AppData\Roaming\Code\User\workspac
45f82063c306\redhat.java\jdt_ws\ai assist_b29d9308\bin' 'ai2'
Original list: [1, 2, 3, 4, 5]
Reversed list: [5, 4, 3, 2, 1]
PS D:\ai assist>

```

Task 3: Few-shot Prompt – String Pattern Matching

Task Description #3

- Write a comment with 2–3 examples to help Copilot understand how to check if a string starts with a capital letter and ends with a period.

Expected Output #3

```
ai3.java > Language Support for Java(TM) by Red Hat > ai3
3 public class ai3 {
    Run main | Debug main | Run | Debug
4     public static void main(String[] args) {
5         String test1 = "Hello world.";
6         String test2 = "hello world.";
7         String test3 = "Hello world";
8
9         System.out.println(checkString(test1)); // true
10        System.out.println(checkString(test2)); // false
11        System.out.println(checkString(test3)); // false
12    }
13
14    public static boolean checkString(String str) {
15        if (str.isEmpty()) {
16            return false;
17        }
18        char firstChar = str.charAt(index: 0);
19        char lastChar = str.charAt(str.length() - 1);
20
21        return Character.isUpperCase(firstChar) && lastChar
22    }
23 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
onMessages' '-cp' 'C:\Users\WINDOWS\AppData\Roaming\Code\User\workspa
45f82063c306\redhat.java\jdt_ws\ai assist_b29d9308\bin' 'ai3'
true
false
false
PS D:\ai assist>
```

Task 4: Zero-shot vs Few-shot – Email Validator

Task Description #4

- First, prompt Copilot to write an email validation function using zero-shot (just the task in comment).
- Then, rewrite the prompt using few-shot examples.

Expected Output #4

The screenshot shows an IDE with a file explorer on the left containing files ai.java, ai2.java, ai3.java, and ai4.java. The main editor displays the code for ai4.java, which includes comments about prompt engineering and a Java class with a main method and a validateEmail method. The terminal at the bottom shows the output of running the program, displaying true, false, and true for the three email addresses.

```
1  /* First, prompt Copilot to write an email validation
2  /* Then, rewrite the prompt using few-shot examples.
3  // Zero-shot prompt to validate an email address
4  import java.util.regex.Pattern;
5  public class ai4 {
6      Run main | Debug main | Run | Debug
7      public static void main(String[] args) {
8          String email1 = "user@example.com";
9          String email2 = "invalid.email";
10         String email3 = "another@domain.org";
11         System.out.println(validateEmail(email1)); // t
12         System.out.println(validateEmail(email2)); // f
13         System.out.println(validateEmail(email3)); // t
14     }
15     public static boolean validateEmail(String email) {
16         String emailRegex = "[a-zA-Z0-9._%+-]+@[a-zA-Z
17         Pattern pattern = Pattern.compile(emailRegex);
18         return pattern.matcher(email).matches();
19     }
20 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
howCodeDetailsInExceptionMessages' '-cp' 'C:\Users\WINDOWS\AppData
ge\272c303df60776de7abd45f82063c306\redhat.java\jdt_ws\ai assist
true
false
true
PS D:\ai assist>
```

Task 5: Prompt Tuning – Summing Digits of a Number

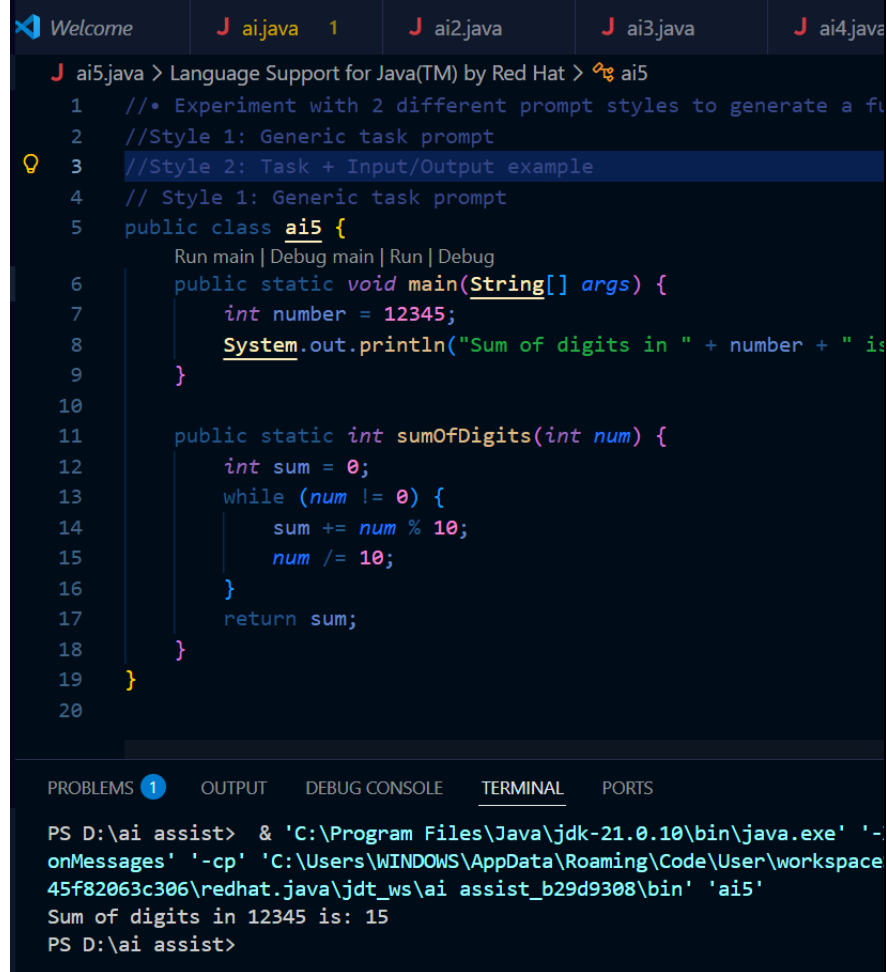
Task Description #5

- Experiment with 2 different prompt styles to generate a function that returns the sum of digits of a number.

Style 1: Generic task prompt

Style 2: Task + Input/Output example

Expected Output #5



The screenshot shows an IDE with a dark theme. The top bar has tabs for 'Welcome', 'ai.java 1', 'ai2.java', 'ai3.java', and 'ai4.java'. The active tab is 'ai5.java', which contains the following code:

```
1  /* Experiment with 2 different prompt styles to generate a fu
2  //Style 1: Generic task prompt
3  //Style 2: Task + Input/Output example
4  // Style 1: Generic task prompt
5  public class ai5 {
6      Run main | Debug main | Run | Debug
7      public static void main(String[] args) {
8          int number = 12345;
9          System.out.println("Sum of digits in " + number + " is
10     }
11     public static int sumOfDigits(int num) {
12         int sum = 0;
13         while (num != 0) {
14             sum += num % 10;
15             num /= 10;
16         }
17         return sum;
18     }
19 }
20
```

Below the code editor is a panel with tabs for 'PROBLEMS 1', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the following command and output:

```
PS D:\ai assist> & 'C:\Program Files\Java\jdk-21.0.10\bin\java.exe' '-
onMessages' '-cp' 'C:\Users\WINDOWS\AppData\Roaming\Code\User\workspace
45f82063c306\redhat.java\jdt_ws\ai assist_b29d9308\bin' 'ai5'
Sum of digits in 12345 is: 15
PS D:\ai assist>
```

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots