

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName: B. Tech		Assignment Type: Lab	AcademicYear: 2025-2026
CourseCoordinatorName		Venkataramana Veeramsetty	
Instructor(s)Name		1. Dr. Mohammed Ali Shaik 2. Dr. T Sampath Kumar 3. Mr. S Naresh Kumar 4. Dr. V. Rajesh 5. Dr. Brij Kishore 6. Dr Pramoda Patro 7. Dr. Venkataramana 8. Dr. Ravi Chander 9. Dr. Jagjeeth Singh	
CourseCode	24CS002PC215	CourseTitle	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment		Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 3.3 (Present assignment number) / 24 (Total number of assignments)			

Q.No.	Question	Expected Time to complete
1	<p>Lab 3: Prompt Engineering – Improving Prompts and Context Management</p> <p>Lab Objectives:</p> <ul style="list-style-type: none"> To understand how prompt structure and wording influence AI-generated code. To explore how context (like comments and function names) helps AI generate relevant output. To evaluate the quality and accuracy of code based on prompt clarity. To develop effective prompting strategies for AI-assisted programming. <p>Lab Outcomes (LOs): After completing this lab, students will be able to:</p> <ul style="list-style-type: none"> Generate Python code using Google Gemini in Google Colab. Analyze the effectiveness of code explanations and suggestions by Gemini. Set up and use Cursor AI for AI-powered coding assistance. Evaluate and refactor code using Cursor AI features. Compare AI tool behavior and code quality across different platforms. 	03.08.2025 EOD

Task Description#1

- Try 3 different prompts to generate a factorial function.

Expected Output#1

- Comparison of AI-generated code styles

Task Description#2

- Provide a clear example input-output prompt to generate a sorting function.

Expected Output#2

- Functional sorting code from AI

Task Description#3

- Start with the vague prompt “Generate python code to calculate power bill” and improve it step-by-step

Expected Output#3

- Enhanced AI output with clearer prompts

Task Description#4

- Write structured comments to help AI generate two linked functions (e.g., login_user() and register_user()).

Expected Output#4

- Consistent functions with shared logic

Task Description#5

- Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

Expected Output#5

- Code quality difference analysis for various prompts

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

Evaluation Criteria:

Criteria	Max Marks
Factorial Function (Task#1)	0.5
Sorting Function (Task#2)	0.5
Vague Vs. Specific Prompting (Task #3)	0.5
Linked Functions (Task #4)	0.5
Temperature Conversion Function (Task #5)	0.5
Total	2.5 Marks

Task Description#1

- Try 3 different prompts to generate a factorial function.

Expected Output#1

- Comparison of AI-generated code styles

Prompt-1:

Generate a function that prints factorial of a given number

Code:

```
task1.py > main
1  def factorial(n):
2      if n < 0:
3          return "Factorial is not defined for negative numbers"
4      elif n == 0 or n == 1:
5          return 1
6      else:
7          result = 1
8          for i in range(2, n + 1):
9              result *= i
10         return result
11
12  def main():
13      Ctrl+L to chat, Ctrl+K to generate
14      try:
15          # Get input from user
16          user_input = input("Enter a number to calculate factorial: ")
17
18          # Convert to integer
19          number = int(user_input)
20
21          # Calculate factorial
22          result = factorial(number)
23
24          # Print the result
25          if isinstance(result, str):
26              print(result)
27          else:
28              print(f"Factorial of {number} is: {result}")
29
30      except ValueError:
31          print("Error: Please enter a valid integer")
32      except Exception as e:
33          print(f"An error occurred: {e}")
34
35  if __name__ == "__main__":
36      main()
37
```

Output:

```
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop/OneDrive/文档/AIAC_LAB_/Lab3/task1.py
Enter a number to calculate factorial: 8
Factorial of 8 is: 40320
```

Prompt 2:

Write a simple python function to find factorial of a number ,collect input from the console

Code:

```
task2.py > ...
1 def factorial(n):
2     if n < 0:
3         return "Error: Factorial is not defined for negative numbers"
4     elif n == 0 or n == 1:
5         return 1
6     else:
7         result = 1
8         for i in range(2, n + 1):
9             result *= i
10        return result
11
12 # Collect input from console
13 user_input = input("Enter a number to calculate factorial: ")
14
15 try:
16     number = int(user_input)
17     result = factorial(number)
18     print(f"Factorial of {number} is: {result}")
19 except ValueError:
20     print("Error: Please enter a valid integer")
21
```

Problems

Output

Debug Console

Terminal

Ports

Python

+

-

□

🗑

...

^

×

```
6
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop/OneDrive/文档/AIAC_LAB_/Lab3/task2.py
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> 
```

Prompt-3:

Create a simple python function that gives factorial of a number,use input from the programmer

```
task2.py > ...
1  def factorial(n):
2      """
3      Calculate the factorial of a given number.
4      """
5      if n < 0:
6          return "Error: Factorial is not defined for negative numbers"
7      elif n == 0 or n == 1:
8          return 1
9      else:
10         result = 1
11         for i in range(2, n + 1):
12             result *= i
13         return result
14
15 # Programmer provides the input number here
16 number = 5 # Change this number to calculate factorial of different numbers
17
18 # Calculate and display the factorial
19 result = factorial(number)
20 print(f"Factorial of {number} is: {result}")
21
```

Problems Output Debug Console **Terminal** Ports

/OneDrive/文档/AIAC_LAB/Lab3/task2.py
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB\Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop/OneDrive/文档/AIAC_LAB/Lab3/task2.py
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB\Lab3>

Task Description#2

- Provide a clear example input-output prompt to generate a sorting function.

Expected Output#2

- Functional sorting code from AI

Prompt:

Generate a python code that takes a input list from the user and sort the list
Ascending order

Code along with output:

```
1 def sort_list_ascending():
2     """
3     Function to take input list from user and sort it in ascending order
4     """
5     try:
6         # Get input from user
7         print("Enter numbers separated by spaces (e.g., 5 2 8 1 9):")
8         user_input = input("Enter your list: ")
9
10        # Convert input string to list of integers
11        numbers = [int(x) for x in user_input.split()]
12
13        # Sort the list in ascending order
14        sorted_list = sorted(numbers)
15
16        # Display the results
17        print(f"Original list: {numbers}")
18        print(f"Sorted list (ascending): {sorted_list}")
19
20    except ValueError:
21        print("Error: Please enter valid numbers separated by spaces")
22    except Exception as e:
23        print(f"An error occurred: {e}")
24
25 def sort_list_with_builtin():
26     """
27     Alternative method using list.sort() method
28     """
29     try:
30         print("Enter numbers separated by spaces:")
31         user_input = input("Enter your list: ")
32
33         # Convert input string to list of integers
34         numbers = [int(x) for x in user_input.split()]
35
36         # Create a copy to avoid modifying original list
37         numbers_copy = numbers.copy()
38
39        # Sort the list in ascending order using sort() method
40        numbers_copy.sort()
41
42        print(f"Original list: {numbers}")
43        print(f"Sorted list (ascending): {numbers_copy}")
44
45    except ValueError:
46        print("Error: Please enter valid numbers separated by spaces")
47
48 # Main execution
49 if __name__ == "__main__":
50     print("=== List Sorting Program ===")
51     print("Method 1: Using sorted() function")
52     sort_list_ascending()
53
54     print("\n" + "-"*40 + "\n")
55
56     print("Method 2: Using sort() method")
57     sort_list_with_builtin()
58
59 Problems Output Debug Console Terminal Ports
Method 1: Using sorted() function
Enter numbers separated by spaces (e.g., 5 2 8 1 9):
Enter your list: 5 66 4 22 3 8 99 0 33
Original list: [5, 66, 4, 22, 3, 8, 99, 0, 33]
sorted list (ascending): [0, 3, 4, 5, 8, 22, 33, 66, 99]
```

Task Description#3

- Start with the vague prompt “Generate python code to
- calculate power bill” and improve it step-by-step

prompt:

generate a python code to calculate powerbill containing previous units
,present units ,read input units from the console

code:

```
task3.py > ...
1 def calculate_power_bill():
2     """
3     Function to calculate power bill based on previous and present units
4     """
5     try:
6         # Get input from user
7         print("=== Power Bill Calculator ===")
8
9         # Get previous units
10        previous_units = float(input("Enter previous month units: "))
11
12        # Get present units
13        present_units = float(input("Enter present month units: "))
14
15        # Calculate units consumed
16        units_consumed = present_units - previous_units
17
18        # Calculate bill based on units consumed
19        if units_consumed <= 0:
20            print("Error: Present units should be greater than previous units")
21            return
22
23        # Define rate structure (example rates - can be modified)
24        if units_consumed <= 100:
25            rate_per_unit = 2.50
26        elif units_consumed <= 200:
27            rate_per_unit = 3.50
28        elif units_consumed <= 300:
29            rate_per_unit = 4.50
30        else:
31            rate_per_unit = 5.50
32
33        # Calculate total bill
34        total_bill = units_consumed * rate_per_unit
35
36        # Display the results
37        print("\n=== Bill Details ===")
38        print(f"Previous month units: {previous_units}")
39        print(f"Present month units: {present_units}")
40        print(f"Units consumed: {units_consumed}")
41        print(f"Rate per unit: ${rate_per_unit}")
42        print(f"Total bill: ${total_bill:.2f}")
43
44    except ValueError:
45        print("Error: Please enter valid numbers")
46    except Exception as e:
47        print(f"An error occurred: {e}")
48
49 def calculate_power_bill_with_tax():
50     """
51     Function to calculate power bill with tax included
52     """
53     try:
54         print("\n=== Power Bill Calculator with Tax ===")
55
56        # Get input from user
57        previous_units = float(input("Enter previous month units: "))
58        present_units = float(input("Enter present month units: "))
59
60        # Calculate units consumed
61        units_consumed = present_units - previous_units
62
63        if units_consumed <= 0:
64            print("Error: Present units should be greater than previous units")
65            return
66
67        # Calculate base bill
68        if units_consumed <= 100:
69            rate_per_unit = 2.50
70        elif units_consumed <= 200:
71            rate_per_unit = 3.50
72        elif units_consumed <= 300:
```

```
73     rate_per_unit = 4.50
74 else:
75     rate_per_unit = 5.50
76
77 base_bill = units_consumed * rate_per_unit
78
79 # Calculate tax (example: 10% tax)
80 tax_rate = 0.10
81 tax_amount = base_bill * tax_rate
82
83 # Calculate total bill with tax
84 total_bill = base_bill + tax_amount
85
86 # Display detailed results
87 print("\n=== Detailed Bill ===")
88 print(f"Previous month units: {previous_units}")
89 print(f"Present month units: {present_units}")
90 print(f"Units consumed: {units_consumed}")
91 print(f"Rate per unit: ${rate_per_unit}")
92 print(f"Base bill: ${base_bill:.2f}")
93 print(f"Tax {(tax_rate*100)%}: ${tax_amount:.2f}")
94 print(f"Total bill: ${total_bill:.2f}")
95
96 except ValueError:
97     print("Error: Please enter valid numbers")
98 except Exception as e:
99     print(f"An error occurred: {e}")
100
101 # Main execution
102 if __name__ == "__main__":
103     # Run basic power bill calculator
104     calculate_power_bill()
105
106     # Run power bill calculator with tax
107     calculate_power_bill_with_tax()
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Task Description#4

- Write structured comments to help AI generate two linked functions (e.g., login_user() and register_user()).

Prompt:

Generate simple two python linked functions one is for register user and login user for account creation

Code:


```

task4.py > ...
1  # Global dictionary to store user accounts
2  user_accounts = {}
3
4  def register_user():
5      """
6      Function to register a new user account
7      """
8      print("=== Account Registration ===")
9
10     # Get user input for account creation
11     username = input("Enter username for your account: ").strip()
12
13     # Check if username already exists
14     if username in user_accounts:
15         print("Error: Username already exists! Please choose a different username.")
16         return False
17
18     # Get password for account
19     password = input("Enter password for your account: ")
20
21     # Store user account credentials
22     user_accounts[username] = password
23
24     print("Account created successfully!")
25     print(f"Username: {username}")
26     print("You can now login with your account.")
27     return True
28
29  def login_user():
30      """
31      Function to login to existing user account
32      """
33      print("=== Account Login ===")
34
35     # Get login credentials
36     username = input("Enter your username: ").strip()
37     password = input("Enter your password: ")
38
39     # Check if account exists and password matches
40     if username in user_accounts and user_accounts[username] == password:
41         print("Login successful!")
42         print(f"Welcome to your account, {username}!")
43         return True
44     else:
45         print("Invalid username or password.")
46         print("Please check your account credentials.")
47         return False
48
49  # Main execution for account management
50  if __name__ == "__main__":
51      while True:
52          print("\n=== Account Management System ===")
53          print("1. Create Account")
54          print("2. Login to Account")
55          print("3. Exit")
56
57          choice = input("Enter your choice (1-3): ").strip()
58
59          if choice == '1':
60              register_user()
61          elif choice == '2':
62              login_user()
63          elif choice == '3':
64              print("Thank you for using Account Management System!")
65              break
66          else:
67              print("Invalid choice! Please try again.")

```

Output:

```
=== User Registration ===
Enter username: 2403a51057
Enter password: hary66
Confirm password: hary66
Enter email: 57@gmail.com
Enter full name: HarichandanaThoopukari
Enter phone number: 1236661230

=== Registration Successful ===
Username: 2403a51057
Email: 57@gmail.com
Full Name: HarichandanaThoopukari
Phone: 1236661230
Registration Date: 2024-01-01
Status: Active

=== User Management System ===
1. Register User
2. Login User
3. Display All Users
4. Exit

Enter your choice (1-4): 2
=== User Login ===
Enter username: 2403a51057
Enter password: hary66

=== Login Successful ===
Welcome back, HarichandanaThoopukari!
Username: 2403a51057
Email: 57@gmail.com
Full Name: HarichandanaThoopukari
Phone: 1236661230
Registration Date: 2024-01-01
Status: Logged In
Last Login: Current Session
```

Task Description#5

- Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

Prompt1:

Write a Python function to convert temperature from Celsius to Fahrenheit or vice-versa as for the user's selection

Code:

```
task5.py > ...
1  def convert_temperature():
2      """
3      Converts temperature between Celsius and Fahrenheit based on user input.
4      """
5      try:
6          print("Temperature Converter")
7          temp = float(input("Enter the temperature value: "))
8          unit = input("Is this in Celsius or Fahrenheit? (C/F): ").strip().upper()
9
10         if unit == 'C':
11             converted = (temp * 9/5) + 32
12             print(f"{temp}°C is {converted:.2f}°F")
13         elif unit == 'F':
14             converted = (temp - 32) * 5/9
15             print(f"{temp}°F is {converted:.2f}°C")
16         else:
17             print("Invalid unit. Please enter 'C' for Celsius or 'F' for Fahrenheit.")
18     except ValueError:
19         print("Invalid input. Please enter a numeric temperature value.")
20
21     # Call the function if this script is run directly
22     if __name__ == "__main__":
23         convert_temperature()
24
```

Problems Output Debug Console Terminal Ports

```
P> 'C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3\task5.py'
Temperature Converter
Enter the temperature value: 15
Is this in Celsius or Fahrenheit? (C/F): c
15.0°C is 59.00°F
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> |
```

Prompt2:

Write a Python function to convert temperature from Celsius to Fahrenheit or vice-versa as for the user's selection.

Code along with output:

task5_1.py > ...

```
1 def convert_temperature():
2     """
3     Converts temperature between Celsius and Fahrenheit based on user's selection.
4     """
5     try:
6         print("Temperature Converter")
7         print("Select conversion type:")
8         print("1. Celsius to Fahrenheit")
9         print("2. Fahrenheit to Celsius")
10        choice = input("Enter 1 or 2: ").strip()
11
12        if choice == '1':
13            celsius = float(input("Enter temperature in Celsius: "))
14            fahrenheit = (celsius * 9/5) + 32
15            print(f"{celsius}°C is {fahrenheit:.2f}°F")
16        elif choice == '2':
17            fahrenheit = float(input("Enter temperature in Fahrenheit: "))
18            celsius = (fahrenheit - 32) * 5/9
19            print(f"{fahrenheit}°F is {celsius:.2f}°C")
20        else:
21            print("Invalid selection. Please enter 1 or 2.")
22    except ValueError:
23        print("Invalid input. Please enter a numeric temperature value.")
24
25 if __name__ == "__main__":
26     convert_temperature()
```

Problems Output Debug Console Terminal Ports

Select conversion type:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter 1 or 2: 1
Enter temperature in Celsius: 35
35.0°C is 95.00°F
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_Lab3>

+ v ...

power
Python
Python