

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		Dr. V. Venkataramana (Co-ordinator)	
		Dr. T. Sampath Kumar	
		Dr. Pramoda Patro	
		Dr. Brij Kishor Tiwari	
		Dr.J.Ravichander	
		Dr. Mohammand Ali Shaik	
		Dr. Anirodh Kumar	
		Mr. S.Naresh Kumar	
		Dr. RAJESH VELPULA	
		Mr. Kundhan Kumar	
		Ms. Ch.Rajitha	
		Mr. M Prakash	
		Mr. B.Raju	
		Intern 1 (Dharma teja)	
		Intern 2 (Sai Prasad)	
		Intern 3 (Sowmya)	
		NS_2 (Mounika)	
CourseCode	24CS002PC215	CourseTitle	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week3 - Wednesday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 6.3(Present assignment number)/24(Total number of assignments)			
Q.No.	Question	Expected Time to complete	
1	<p>Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals</p> <p>Lab Objectives:</p> <ul style="list-style-type: none"> To explore AI-powered auto-completion features for core Python constructs. To analyze how AI suggests logic for class definitions, loops, and conditionals. To evaluate the completeness and correctness of code generated by AI assistants. <p>Lab Outcomes (LOs):</p>	Week3 - Wednesday	

After completing this lab, students will be able to:

- Use AI tools to generate and complete class definitions and methods.
- Understand and assess AI-suggested loops for iterative tasks.
- Generate conditional statements through prompt-driven suggestions.
- Critically evaluate AI-assisted code for correctness and clarity.

Task Description#1 (Classes)

- Use AI to complete a Student class with attributes and a method.
- Check output
- Analyze the code generated by AI tool

Instructions:

- **Initialize class with attributes like name, roll no, marks**
- **Method to display student details**
- **Method to calculate grade based on marks (A: >=90, B: >=75, C: >=60, else Fail)**

Start Writing code and auto complete using any AI tool

Expected Output#1

- Class with constructor and display_details() method

CODE:

```
Task1.py Lab-6 X Task2.py Lab-6 X task1_1.py task1_2.py task1_3.py
Lab-6 > Task1.py > ...
1 class Student:
2     def __init__(self, name, rollno, marks):
3         self.name = name
4         self.rollno = rollno
5         self.marks = marks
6     def display_details(self):
7         print(f"Name: {self.name}, Roll: {self.rollno}, Marks: {self.marks}")
8     def calculate_grade(self):
9         if self.marks >= 90: return "A"
10        elif self.marks >= 75: return "B"
11        elif self.marks >= 60: return "C"
12        else: return "Fail"
13    name = input("Enter name: ")
14    rollno = int(input("Enter roll no: "))
15    marks = int(input("Enter marks: "))
16    student = Student(name, rollno, marks)
17    student.display_details()
18    print(f"Grade: {student.calculate_grade()}")
```

OUTPUT:

```
Problems Output Debug Console Terminal Ports
Enter roll no: 1121
Enter marks: 96
Name: deeksha, Roll: 1121, Marks: 96
Grade: A
PS C:\Users\DEEKSHA\OneDrive\Desktop\AIAC> 
```

Explanation:

1. **Class** :Creates a Student class with constructor that takes name, roll number, and marks as parameters.

2. `display_details()` prints the student's information in a formatted string.
3. `calculate_grade()` returns letter grades based on marks: A (≥ 90), B (≥ 75), C (≥ 60), or "Fail" (< 60).
4. user to enter student details (name, roll number, marks) and converts roll number and marks to integers.
5. Creates a Student object, displays the student details, and prints the calculated grade.

Task Description#2 (Loops)

- Prompt AI to complete a function that prints the first 10 multiples of a number using a loop.
- Analyze the generated code
- Ask AI to generate code using other controlled looping

Write code using **For** Loop, later complete code using **While** Loop

Expected Output#2

- Correct loop-based implementation

CODE :

```
Lab-6 > Task2.py > ...
1  def print_multiples_for(number):
2      print("Using FOR loop:")
3      for i in range(1, 11):
4          result = number * i
5          print(f"{number} x {i} = {result}")
6  def print_multiples_while(number):
7      print("\nUsing WHILE loop:")
8      i = 1
9      while i <= 10:
10         result = number * i
11         print(f"{number} x {i} = {result}")
12         i = i + 1
13  num = int(input("Enter Number to see its first 10 multiples: "))
14  print_multiples_for(num)
15  print_multiples_while(num)
16
```

OUTPUT:

```
1.exe c:/Users/DEEKSHA/OneDrive/Desktop/AIAC/Lab-6/Task2.py
Enter Number to see its first 10 multiples: 5
Using FOR loop:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Using WHILE loop:

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

```
PS C:\Users\DEEKSHA\OneDrive\Desktop\AIAC>
```

Explanation:

1. **Two Functions:** One uses FOR loop, other uses WHILE loop to print multiplication tables
2. **FOR Loop:** Uses range(1, 11) to iterate 1-10 times
3. **WHILE Loop:** Uses counter variable i that increments from 1 to 10
4. Takes a number and shows its first 10 multiples
5. Displays same multiplication table twice using different loop methods

Both functions produce identical results showing 1×num through 10×num.

Task Description#3 (Conditional Statements)

- Ask AI to write nested if-elif-else conditionals to classify age groups.
- Analyze the generated code
- Ask AI to generate code using other conditional statements

Table: Age Group Classification Logic

Age Range	Age Group
0 – 12 years	Child
13 – 19 years	Teen
20 – 59 years	Adult
60 years & above	Senior

•
Expected Output#3

- Age classification function with appropriate conditions and with explanation

CODE :

```

Lab-6 > Task3.py > age
1 age = int(input("Enter your age: "))
2
3 # Nested if-elif-else conditionals to classify age groups
4 if age >= 0:
5     if age <= 12:
6         print(f"Age {age}: Child")
7     elif age <= 19:
8         print(f"Age {age}: Teen")
9     elif age <= 59:
10        print(f"Age {age}: Adult")
11    else:
12        print(f"Age {age}: Senior")
13 else:
14    print(f"Age {age}: Invalid age (must be 0 or positive)")

```

OUTPUT:

```

Enter your age: 13
Age 13: Teen

```

```

Enter your age: 11
Age 11: Child

```

```

Enter your age: 45
Age 45: Adult

```

```

Enter your age: 61
Age 61: Senior

```

Explanation:

1. Takes age as integer input from the user
2. Nested Structure: Uses nested if-elif-else statements to categorize age groups
3. Age Categories: Child (0-12), Teen (13-19), Adult (20-59), Senior (60+)
4. Validation: Checks if age is non-negative (≥ 0) before classification
5. Output: Prints the age with its corresponding category, or "Invalid age" for negative values

Task Description#4 (For and While loops)

- Generate a sum_to_n() function to calculate sum of first n numbers
- Analyze the generated code
- Get suggestions from AI with other controlled looping

Expected Output#4

- Python code with explanation

CODE:

```

Lab-6 > Task4.py > n
1  n = int(input("Enter a number: "))
2
3  # Method 1: Using for loop
4  total_for = 0
5  for i in range(1, n + 1):
6      total_for = total_for + i
7
8  # Method 2: Using while loop
9  total_while = 0
10 i = 1
11 while i <= n:
12     total_while = total_while + i
13     i = i + 1
14
15 # Show results
16 print(f"Sum of first {n} numbers:")
17 print(f"For loop: {total_for}")
18 print(f"While loop: {total_while}")

```

OUTPUT:

```

1.exe c:/Users/DEEKSHA/OneDrive/Desktop/AIAC/Lab-6/Task4.py
Enter a number: 5
Sum of first 5 numbers:
For loop: 15
While loop: 15

```

Explanation:

Sum of First N Numbers:

1. Takes number n from user
2. FOR Loop: Adds 1 to n using range(1, n+1)
3. WHILE Loop: Adds 1 to n using counter variable
4. Same Result: Both methods calculate 1+2+3+...+n
5. Output: Shows sum from both loop types

Two different loops, same calculation.

Task Description#5 (Class)

- Use AI to build a BankAccount class with deposit, withdraw, and balance methods.
- Analyze the generated code
- Add comments and explain code

Instructions

- Initialize BankAccount class with attributes like name, balance
- Method to deposit amount
- Method to withdraw amount
- Method to check balance

Expected Output#5

- Python code with explanation

CODE :

```

Lab-6 > Task5.py > BankAccount > deposit
1 class BankAccount:
2     def __init__(self, name, balance=0):
3         self.name = name
4         self.balance = balance
5     def deposit(self, amount):
6         if amount > 0:
7             self.balance += amount
8             print(f"₹{amount} deposited. New balance: ₹{self.balance}")
9         else:
10            print("Deposit amount must be positive.")
11    def withdraw(self, amount):
12        if amount > self.balance:
13            print("Insufficient balance.")
14        elif amount <= 0:
15            print("Withdrawal amount must be positive.")
16        else:
17            self.balance -= amount
18            print(f"₹{amount} withdrawn. New balance: ₹{self.balance}")
19    def check_balance(self):
20        print(f"Current balance: ₹{self.balance}")
21    # Example usage inside the Bank_Account function:
22    name = input("Enter account holder name: ")
23    account = BankAccount(name)

```

```

while True:
    choice = input("Enter your choice (1-4): ")
    if choice == "1":
        amt = float(input("Enter amount to deposit: "))
        account.deposit(amt)
    elif choice == "2":
        amt = float(input("Enter amount to withdraw: "))
        account.withdraw(amt)
    elif choice == "3":
        account.check_balance()
    elif choice == "4":
        print("Thank you for using the Bank Account!")
        break
    else:
        print("Invalid choice. Please try again.")

```

OUTPUT:

```

PS C:\Users\DEEKSHA\OneDrive\Desktop\AIAC> & C:/Users/DEEKSHA/AppData/Local/
DEEKSHA/OneDrive/Desktop/AIAC/Lab-6/Task5.py
Enter account holder name: deeksha
Enter your choice (1-4): 1
Enter amount to deposit: 10000
₹10000.0 deposited. New balance: ₹10000.0
Enter your choice (1-4): 2
Enter amount to withdraw: 2000
₹2000.0 withdrawn. New balance: ₹8000.0
Enter your choice (1-4): 3
Current balance: ₹8000.0
Current balance: ₹8000.0
Enter your choice (1-4): 4
Thank you for using the Bank Account!

```

Explanation:

1. **Class Definition:** BankAccount class with constructor taking name and optional balance (default 0)
2. **Methods:** deposit() (validates positive amount), withdraw() (checks sufficient

- balance), check_balance() (displays current balance)
3. **User Input:** Takes account holder name and creates account object
 4. **Menu Loop:** Infinite while loop with 4 options: deposit, withdraw, check balance, or exit
 5. **Validation:** All methods include input validation (positive amounts, sufficient funds) with appropriate error messages

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
Class	1.0
Loops	1.0
Conditional Statements	0.5
Total	2.5 Marks