

Lab Assignment 6.3 – AI Assisted Coding

Question 1: Student Class

Algorithm:

1. Define a Student class
2. Initialize attributes using constructor
3. Create display method
4. Create object and display details

Pseudo Code:

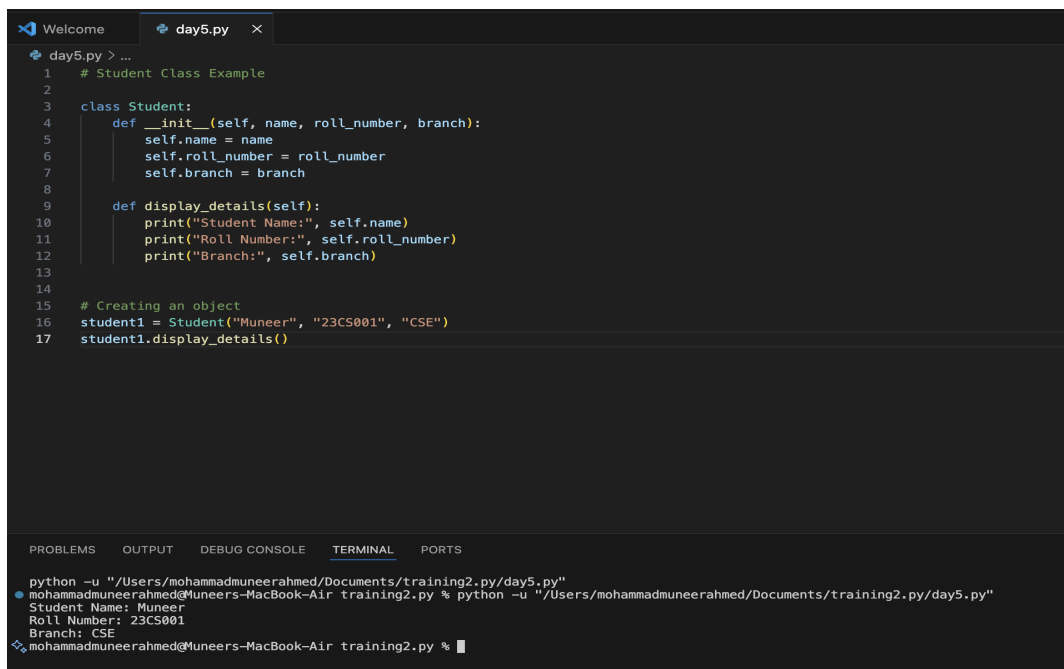
START

Define class Student

Initialize name, roll, branch

Display details

END



```
day5.py > ...
1  # Student Class Example
2
3  class Student:
4      def __init__(self, name, roll_number, branch):
5          self.name = name
6          self.roll_number = roll_number
7          self.branch = branch
8
9      def display_details(self):
10         print("Student Name:", self.name)
11         print("Roll Number:", self.roll_number)
12         print("Branch:", self.branch)
13
14
15 # Creating an object
16 student1 = Student("Muneer", "23CS001", "CSE")
17 student1.display_details()
```

python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"

Student Name: Muneer
Roll Number: 23CS001
Branch: CSE

Question 2: Multiples Using Loops

Algorithm:

1. Take a number
2. Use loop to print first 10 multiples

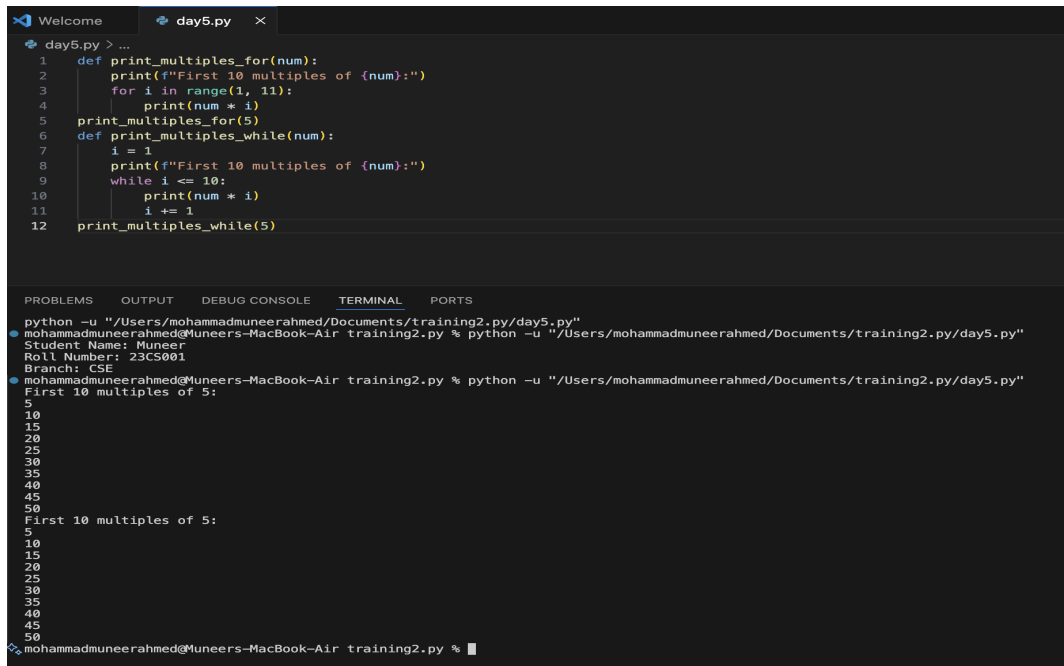
Pseudo Code:

START

FOR i from 1 to 10

Print num * i

END



```
day5.py > ...
1 def print_multiples_for(num):
2     print(f"First 10 multiples of {num}:")
3     for i in range(1, 11):
4         print(num * i)
5 print_multiples_for(5)
6 def print_multiples_while(num):
7     i = 1
8     print(f"First 10 multiples of {num}:")
9     while i <= 10:
10        print(num * i)
11        i += 1
12 print_multiples_while(5)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
Student Name: Muneer
Roll Number: 23CS001
Branch: CSE
mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
First 10 multiples of 5:
5
10
15
20
25
30
35
40
45
50
First 10 multiples of 5:
5
10
15
20
25
30
35
40
45
50
mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```

Question 3: Age Classification

Algorithm:

1. Take age as input
2. Use if-elif-else to classify

Pseudo Code:

START

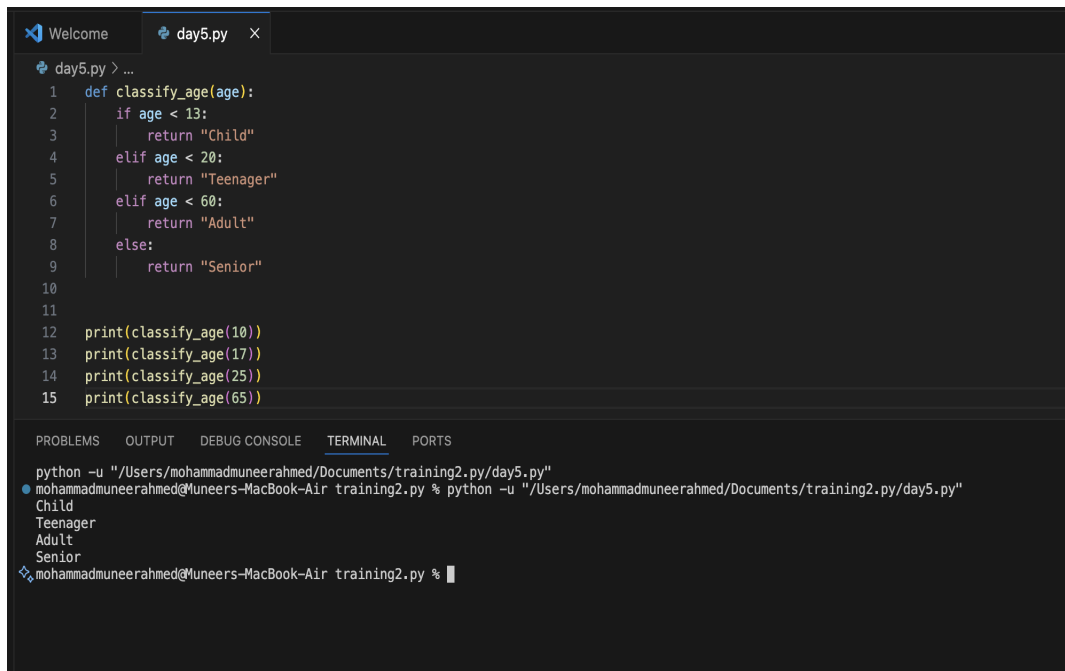
IF age < 13 print Child

ELSE IF age < 20 print Teenager

ELSE IF age < 60 print Adult

ELSE print Senior

END



```
day5.py > ...
1 def classify_age(age):
2     if age < 13:
3         return "Child"
4     elif age < 20:
5         return "Teenager"
6     elif age < 60:
7         return "Adult"
8     else:
9         return "Senior"
10
11
12 print(classify_age(10))
13 print(classify_age(17))
14 print(classify_age(25))
15 print(classify_age(65))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
● mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
Child
Teenager
Adult
Senior
❖ mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```

Question 4: Sum of First n Numbers

Algorithm:

1. Initialize sum
2. Use loop/formula to compute sum

Pseudo Code:

START

sum = $n \cdot (n+1) / 2$

Print sum

END

```
day5.py > ...
1  def sum_to_n_for(n):
2      total = 0
3      for i in range(1, n + 1):
4          total += i
5      return total
6  print(sum_to_n_for(10))
7  def sum_to_n_while(n):
8      total = 0
9      i = 1
10     while i <= n:
11         total += i
12         i += 1
13     return total
14
15
16  print(sum_to_n_while(10))
17  def sum_to_n_formula(n):
18      return n * (n + 1) // 2
19
20
21  print(sum_to_n_formula(10))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
55
55
mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```

Question 5: Bank Account Class

Algorithm:

1. Define BankAccount class
2. Implement deposit, withdraw, balance methods

Pseudo Code:

START

Create class BankAccount

Deposit, Withdraw, Check balance

END

```
day5.py > BankAccount > withdraw
1  # Bank Account Class
2  class BankAccount:
3      def __init__(self, account_holder, balance=0):
4          self.account_holder = account_holder
5          self.balance = balance
6
7      def deposit(self, amount):
8          if amount > 0:
9              self.balance += amount
10             print(f"Deposited ₹{amount}")
11         else:
12             print("Invalid deposit amount")
13
14     def withdraw(self, amount):
15         if amount <= self.balance:
16             self.balance -= amount
17             print(f"Withdrawn ₹{amount}")
18         else:
19             print("Insufficient balance")
20
21     def check_balance(self):
22         print(f"Current Balance: ₹{self.balance}")
23
24
25 # Testing the class
26 account = BankAccount("Muneer", 1000)
27 account.check_balance()
28 account.deposit(500)
29 account.withdraw(300)
30 account.check_balance()
```

python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"

Current Balance: ₹1000
Deposited ₹500
Withdrawn ₹300
Current Balance: ₹1200

mohammadmuneerahmed@Muneers-MacBook-Air training2.py %