

AI ASSISTED CODING

Hall ticket :2403A51270

Batch: 12

GENERATED BY GITHUB COPILOT..

TASK-1 :

Code:

```

Users > DELL > Documents > .vscode > jh.py > ...
# BankAccount class represents a simple bank account with deposit, withdraw, and balance display

class BankAccount:
    def __init__(self, account_holder, balance=0):
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited ${amount}. New balance: ${self.balance}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if amount > 0:
            if amount <= self.balance:
                self.balance -= amount
                print(f"Withdrew ${amount}. New balance: ${self.balance}")
            else:
                print("Insufficient funds.")
        else:
            print("Withdrawal amount must be positive.")

    def display_balance(self):
        print(f"{self.account_holder}'s balance: ${self.balance}")

account = BankAccount("Alice", 100)
account.display_balance()
account.deposit(50)
account.withdraw(30)
account.withdraw(150)
account.display_balance()

```

Output:

```

Alice's balance: $120
PS C:\Users\DELL\profile wtmp> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe
ments/.vscode/jh.py
Alice's balance: $100
Deposited $50. New balance: $150
Withdrew $30. New balance: $120
Insufficient funds.
Alice's balance: $120
PS C:\Users\DELL\profile wtmp>

```

TASK-2:

Code:

```
> Users > DELL > Documents > .vscode > ww.py > ...
1  # Loop through the list and sum all even numbers
2
3  numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
4  even_sum = 0
5  for num in numbers:
6      if num % 2 == 0:
7          even_sum += num
8
9  print(f"Sum of even numbers: {even_sum}")
```

Output:

```
PS C:\Users\DELL\profile wtmp> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe
ments/.vscode/ww.py
Sum of even numbers: 30
PS C:\Users\DELL\profile wtmp>
```

TASK-3

Code:

```
# Function to determine age group based on age

def age_group(age):
    if age < 13:
        return "Child"
    elif age < 20:
        return "Teenager"
    elif age < 60:
        return "Adult"
    else:
        return "Senior"

# Sample usage
print(age_group(45)) # Output:
```

Output:

```
PS C:\Users\DELL\profile wtmp> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe
ments/.vscode/wmb.py
Adult
PS C:\Users\DELL\profile wtmp>
```

TASK-4:

CODE:

```
1  # While loop to reverse the digits of a number
2
3  num = 1234
4  reversed_num = 0
5  while num > 0:
6      digit = num % 10
7      reversed_num = reversed_num * 10 + digit
8      num //= 10
9
10 print(reversed_num)  # Output: 4321
```

OUTPUT:

```
PS C:\Users\DELL\profile wtmp> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe
ments/.vscode/wmb.py
4321
PS C:\Users\DELL\profile wtmp>
```

TASK-5:

CODE:

```
# Employee base class and Manager derived class with constructor chaining

class Employee:
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary

    def display(self):
        print(f"Name: {self.name}, Salary: {self.salary}")

class Manager(Employee):
    def __init__(self, name, salary, department):
        super().__init__(name, salary)
        self.department = department

    def display(self):
        print(f"Name: {self.name}, Salary: {self.salary}, Dept: {self.department}")

# Sample usage
mgr = Manager("John", 50000, "IT")
mgr.display() # Output: Name: John,
```

OUTPUT:

```
PS C:\Users\DELL\profile wtmp> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe
ments/.vscode/wmb.py
Name: John, Salary: 50000, Dept: IT
PS C:\Users\DELL\profile wtmp>
```