Lab-9 Submission

# NAME : ALUWALA SHARANYA

# HALTICKET\_NO : 2403A52013

# BATCH\_NO : 02(CSE-AIML)

**Task 1: Discount Calculator**

**Question:** Write a Python program to calculate the price after applying discount.

Code :

# task1\_calculate\_discount.py

def calculate\_discount\_original(price, discount\_rate):

# Original function (no validation)

return price - (price \* discount\_rate / 100)

def calculate\_discount\_ai(price, discount\_rate):

# Auto-generated style comments (example)

# Validate input values to avoid wrong calculation

if price < 0:

raise ValueError("price must be >= 0")

if not (0 <= discount\_rate <= 100):

raise ValueError("discount\_rate must be between 0 and 100")

# Calculate discount amount and subtract from price

discount\_amount = price \* discount\_rate / 100

return price - discount\_amount

def calculate\_discount\_final(price, discount\_rate):

"""Calculate the price after applying a percentage discount.

Args:

price (float): Original price (must be >= 0).

discount\_rate (float): Discount percent (0 - 100).

Returns:

float: Price after discount applied.

Raises:

ValueError: If inputs are invalid.

"""

# Input validation (avoid silent errors)

if price < 0:

raise ValueError("price must be >= 0")

if not (0 <= discount\_rate <= 100):

raise ValueError("discount\_rate must be between 0 and 100")

# Compute discount amount and return final price

discount\_amount = price \* discount\_rate / 100

return price - discount\_amount

print(calculate\_discount\_final(200, 10))

if \_\_name\_\_ == "\_\_main\_\_":

price = float(input("Enter price: "))

discount = float(input("Enter discount rate (%): "))

final\_price = calculate\_discount\_final(price, discount)

print("Final price after discount:", final\_price)

### Task 2: Library Management System

**Question:** Create functions to add books and issue books in a library.

Code :

def add\_book(title, author, year):

"""Add a new book record to the library.

Args:

title (str): Book title.

author (str): Author name.

year (int): Year of publication.

Returns:

dict: Book details with id.

"""

book\_id = 1

return {"book\_id": book\_id, "title": title, "author": author, "year": year}

def issue\_book(book\_id, user\_id):

"""Issue a book to a user.

Args:

book\_id (int): ID of the book.

user\_id (int): ID of the user.

Returns:

bool: True if issued successfully.

"""

return True

if \_\_name\_\_ == "\_\_main\_\_":

book = add\_book("Python Basics", "Guido", 2020)

print("Book Added:", book)

issued = issue\_book(book["book\_id"], 101)

print("Book Issued:", issued)

**Output:**

Book Added: {'book\_id': 1, 'title': 'Python Basics', 'author': 'Guido', 'year': 2020}

Book Issued: True

### Task 3: Student Grade System

**Question:** Write a Python function to calculate grade based on marks.

Code :

def calculate\_grade(marks):

"""Calculate grade based on marks.

Args:

marks (int): Marks scored by student.

Returns:

str: Grade (A, B, C, D, F).

"""

if marks >= 90:

return "A"

elif marks >= 75:

return "B"

elif marks >= 60:

return "C"

elif marks >= 40:

return "D"

else:

return "F"

if \_\_name\_\_ == "\_\_main\_\_":

print("Grade for 95:", calculate\_grade(95))

print("Grade for 72:", calculate\_grade(72))

print("Grade for 50:", calculate\_grade(50))

print("Grade for 30:", calculate\_grade(30))

**Output:**

Grade for 95: A

Grade for 72: B

Grade for 50: C

Grade for 30: F

### Task 4: Student Management System

**Question:** Write a Python program to manage student records (add, display, search)

Code :

students = [] # list to store student records

def add\_student(roll\_no, name, marks):

"""Add new student to the system"""

student = {"roll\_no": roll\_no, "name": name, "marks": marks}

students.append(student)

def display\_students():

"""Display all students"""

if not students:

print("No students found.")

else:

for s in students:

print(f"Roll No: {s['roll\_no']}, Name: {s['name']}, Marks: {s['marks']}")

def search\_student(roll\_no):

"""Search student by roll number"""

for s in students:

if s["roll\_no"] == roll\_no:

return s

return None

if \_\_name\_\_ == "\_\_main\_\_":

# sample data

add\_student(1, "Ravi", 85)

add\_student(2, "Priya", 92)

add\_student(3, "Arjun", 60)

print("\nAll Students:")

display\_students()

print("\nSearching Roll No 2:")

result = search\_student(2)

if result:

print(f"Found: {result}")

else:

print("Student not found")

**Output:**

All Students:

Roll No: 1, Name: Ravi, Marks: 85

Roll No: 2, Name: Priya, Marks: 92

Roll No: 3, Name: Arjun, Marks: 60

Searching Roll No 2:

Found: {'roll\_no': 2, 'name': 'Priya', 'marks': 92}



