Decision Making – Labs

Lab 1: Program to check if a number is even or odd

Solution:

```
# Input: Take a number from the user
number = int(input("Enter a number: "))

# Check if the number is even or odd using a single if condition
if number % 2 == 0:
    print("Even")
else:
    print("Odd")
```

Lab 2: Program to accept name and salary. Check if their salary is >3L and display if they have to pay tax

```
# Accept employee details
name = input("Enter employee's name: ")
salary = float(input("Enter employee's annual salary: "))

# Check if salary is greater than 3,00,000
if salary > 300000:
    print(f"{name} has an annual salary of ₹{salary}.")
    print(f"{name} has to pay taxes.")
else:
    print(f"{name} has an annual salary of ₹{salary}.")
    print(f"{name} does not have to pay taxes.")
```

Lab 3: To find the largest of 3 numbers

Solution:

```
# Input three numbers
num1 = 10
num2 = 25
num3 = 15

# Find the largest
if num1 >= num2 and num1 >= num3:
    print("The largest number is:", num1)
elif num2 >= num1 and num2 >= num3:
    print("The largest number is:", num2)
else:
    print("The largest number is:", num3)
```

Lab 4: Program to check if a year given is a leap year or not

Solution:

```
year = 2024
if year % 4 == 0:
    if year % 100 == 0:
        if year % 400 == 0:
            print("Leap year")
        else:
            print("Not a leap year")
    else:
        print("Leap year")
else:
    print("Not a leap year")
```

Lab 5: Student Report Card Problem

Write a program to accept a student’s name and scores in three subjects. Display the total, average, and class secured based on the following criteria:

- **1st Class**: Average score of 60 and above.
- **2nd Class**: Average score of 50 and above.
- **Pass Class**: Average score of 35 and above.
- **Fail**: Average score less than 35.

Solution:

```
# Input student name and scores
name = input("Enter the student's name: ")
score1 = float(input("Enter score for subject 1: "))
score2 = float(input("Enter score for subject 2: "))
score3 = float(input("Enter score for subject 3: "))

# Check if the student passed in each subject
if score1 < 35 or score2 < 35 or score3 < 35:
    result = "Fail"
else:
    # Calculate total and average
    total = score1 + score2 + score3
    average = total / 3

    # Determine the class
    if average >= 60:
        result = "1st Class"
    elif average >= 50:
        result = "2nd Class"
    elif average >= 35:
        result = "Pass Class"
    else:
        result = "Fail"

# Display the results
print("\nStudent Report Card")
print("Name:", name)
if result == "Fail":
    print("Result:", result)
else:
    print("Total:", total)
    print("Average:", average)
    print("Result:", result)
```

Lab 6: Tax Calculator Problem

GlobalNext Solutions, a rapidly growing IT company, employs a diverse workforce ranging from entry-level developers to senior executives. The HR department wants to streamline the tax calculation process for employees under the New Tax Regime (2023). They've decided to build a tax calculation program that computes salaries, taxes, and net incomes while ensuring compliance with the latest tax laws.

As a software developer in GlobalNext's HR-Tech team, you are tasked with developing this program. The system should process employee salary details, validate inputs, calculate taxes, and generate detailed reports.

The program should:

1. Accept employee details, including monthly salary components.
2. Calculate gross and taxable income according to the New Tax Regime (2023).
3. Compute the tax payable using the appropriate tax slabs.
4. Apply any applicable standard deductions and rebates.
5. Generate reports detailing gross salary, taxable income, tax payable, and net salary.

Level 1: Basic Input and Salary Calculation

Objective: Capture employee details and calculate the gross salary.

Tasks:

- Accept the following inputs for an employee:
  - Name
  - EmpID
  - Basic Monthly Salary
  - Special Allowances (Monthly)
  - Bonus Percentage (Annual Bonus as % of Gross Salary)
- Calculate:
  - **Gross Monthly Salary** = Basic Salary + Special Allowances
  - **Annual Gross Salary** = (Gross Monthly Salary × 12) + Bonus
- **Output:**
  - Display the employee details, gross monthly salary, and annual gross salary.

Solution:

```
# Accepting employee details
name = input("Enter employee's name: ")
emp_id = input("Enter employee ID: ")
basic_salary = float(input("Enter basic monthly salary: "))
special_allowances = float(input("Enter monthly special allowances: "))
bonus_percentage = float(input("Enter annual bonus percentage (as a % of gross salary): "))

# Calculate gross monthly salary
gross_monthly_salary = basic_salary + special_allowances

# Calculate annual gross salary
annual_gross_salary = (gross_monthly_salary * 12) + (gross_monthly_salary * bonus_percentage / 100)
```

```
# Output the details
print("\nEmployee Details:")
print(f"Name: {name}")
print(f"Employee ID: {emp_id}")
print(f"Gross Monthly Salary: ₹{gross_monthly_salary:,.2f}")
print(f"Annual Gross Salary: ₹{annual_gross_salary:,.2f}")
```

Level 2: Taxable Income Calculation

Objective: Calculate taxable income after standard deductions.

Tasks:

- Deduct a **Standard Deduction of ₹50,000** from the annual gross salary.
- Compute the **Taxable Income** and display all intermediate calculations.

Output: Display gross salary, standard deduction and taxable income.

Solution:

```
# Define the standard deduction amount
standard_deduction = 50000

# Calculate the taxable income after standard deduction
taxable_income = annual_gross_salary- standard_deduction

# Output the details
print("\nTaxable Income Calculation:")
print(f"Annual Gross Salary: ₹{annual_gross_salary:,.2f}")
print(f"Standard Deduction: ₹{standard_deduction:,.2f}")
print(f"Taxable Income: ₹{taxable_income:,.2f}")
```

Level 3: Tax and Rebate Calculation

Objective: Compute tax payable using the **New Tax Regime (2023)** slabs.

Tasks:

1. Calculate tax based on the following slabs:
   - ₹0- ₹3,00,000: 0%
   - ₹3,00,001- ₹6,00,000: 5%
   - ₹6,00,001- ₹9,00,000: 10%
   - ₹9,00,001- ₹12,00,000: 15%
   - ₹12,00,001- ₹15,00,000: 20%
   - Above ₹15,00,000: 30%
2. Apply **Section 87A Rebate**:
   - Taxable income ≤ ₹7,00,000 → 100% rebate (tax payable = ₹0).
3. Add a **4% Health and Education Cess** to the calculated tax.

Output:

- Display a detailed tax breakdown, including slabs, cess, and total tax payable.

Solution:

```
# Input taxable income
taxable_income = float(input("Enter taxable income (₹): "))

# Initialize variables
tax = 0

# Calculate tax based on slabs
if taxable_income > 1500000:
    tax += (taxable_income- 1500000) * 0.30
    taxable_income = 1500000
if taxable_income > 1200000:
    tax += (taxable_income- 1200000) * 0.20
    taxable_income = 1200000
if taxable_income > 900000:
    tax += (taxable_income- 900000) * 0.15
    taxable_income = 900000
if taxable_income > 600000:
    tax += (taxable_income- 600000) * 0.10
    taxable_income = 600000
if taxable_income > 300000:
    tax += (taxable_income- 300000) * 0.05

# Apply Section 87A rebate
if taxable_income <= 700000:
    tax = 0

# Add 4% Health and Education Cess
cess = tax * 0.04
total_tax_payable = tax + cess

# Display detailed tax breakdown
print("\n--- Tax Breakdown---")
print(f"Base Tax: ₹{tax:.2f}")
print(f"Health and Education Cess (4%): ₹{cess:.2f}")
print(f"Total Tax Payable: ₹{total_tax_payable:.2f}")
```

Level 4: Net Salary Calculation

Objective: Calculate annual net salary after tax deductions.

Tasks:

1. Compute Net Salary = Annual Gross Salary- Total Tax Payable.
2. Display:
   - Annual Gross Salary
   - Total Tax Payable (including cess)
   - Annual Net Salary

Solution:

```
# Compute Net Salary
annual_net_salary = annual_gross_salary- total_tax_payable

# Display results
print("\n--- Net Salary Details---")
print(f"Annual Gross Salary: ₹{annual_gross_salary:.2f}")
print(f"Total Tax Payable: ₹{total_tax_payable:.2f}")
print(f"Annual Net Salary: ₹{annual_net_salary:.2f}")
```

Level 5: Report Generation

Objective: Generate a detailed report for employees.

Tasks:

1. Summarize all computed details:
   - Employee Details (Name, EmpID)
   - Gross Monthly Salary
   - Annual Gross Salary
   - Taxable Income
   - Tax Payable (with breakdown)
   - Annual Net Salary
2. Format the output as a report for better readability.

Output:

- Provide a clean, tabular report for employees.

Example Output (For Reports Level)

Employee Tax Report

Field	Details
Name	John Doe
EmpID	E12345
Gross Monthly Salary	₹85,000
Annual Gross Salary	₹10,20,000
Taxable Income	₹9,70,000
Tax Payable	₹76,800
Annual Net Salary	₹9,43,200

Solution:

```
# Display the report
print("\n--- Employee Tax Report---")
print(f"{'Field':<25}{'Details':<20}")
print("-" * 45)
print(f"{'Name':<25}{employee_name:<20}")
print(f"{'EmpID':<25}{employee_id:<20}")
print(f"{'Gross Monthly Salary':<25}₹{gross_monthly_salary:,.2f}")
print(f"{'Annual Gross Salary':<25}₹{annual_gross_salary:,.2f}")
print(f"{'Taxable Income':<25}₹{taxable_income:,.2f}")
print(f"{'Tax Payable':<25}₹{total_tax_payable:,.2f}")
print(f"{'Annual Net Salary':<25}₹{annual_net_salary:,.2f}")
```

Level 6: Input Validation Rules

Objective: Validate all inputs to ensure accuracy and correctness.

Validation Rules:

1. **Employee Details:**
   - Name: Non-empty, alphabets only, max 50 characters.
   - EmpID: Alphanumeric, 5–10 characters.
2. **Salary Inputs:**
   - Basic Salary: Positive number, max ₹1,00,00,000.
   - Special Allowances: Non-negative, max ₹1,00,00,000.
   - Bonus Percentage: Numeric value, 0–100.
3. **Derived Calculations:**
   - Gross Monthly Salary must be greater than zero.
   - Annual Gross Salary should not exceed realistic values.
4. **General:**
   - Reject invalid inputs with a clear error message.
   - Provide re-entry prompts for invalid data.

Output:

- Indicate if any inputs are invalid and prompt for correction.

Solution:

```
# Input validation for employee details
while True:
    name = input("Enter employee name: ")
    if not name or not name.isalpha() or len(name) > 50:
        print("Error: Name must be non-empty, contain only alphabets, and be at most 50 characters
long.")
    else:
        break

while True:
    emp_id = input("Enter employee ID: ")
    if not emp_id.isalnum() or not (5 <= len(emp_id) <= 10):
        print("Error: Employee ID must be alphanumeric and 5–10 characters long.")
    else:
        break
# Input validation for salary
while True:
    try:
        basic_salary = float(input("Enter basic monthly salary: "))
        if basic_salary <= 0 or basic_salary > 10000000:
            print("Error: Basic salary must be a positive number and not exceed ₹1,00,00,000.")
        else:
            break
    except ValueError:
        print("Error: Please enter a valid number.")

# Input validation for special allowances
while True:
    try:
        special_allowances = float(input("Enter special allowances: "))
        if special_allowances < 0 or special_allowances > 10000000:
            print("Error: Special allowances must be non-negative and not exceed ₹1,00,00,000.")
        else:
            break
    except ValueError:
        print("Error: Please enter a valid number.")

# Input validation for bonus percentage
while True:
    try:
        bonus_percentage = float(input("Enter annual bonus percentage: "))
        if not (0 <= bonus_percentage <= 100):
            print("Error: Bonus percentage must be between 0 and 100.")
        else:
            break
    except ValueError:
        print("Error: Please enter a valid number.")

# Derived calculations validation
gross_monthly_salary = basic_salary + special_allowances
if gross_monthly_salary <= 0:
    print("Error: Gross monthly salary must be greater than zero.")

annual_gross_salary = (gross_monthly_salary * 12) + ((gross_monthly_salary * bonus_percentage) / 100)
if annual_gross_salary > 100000000:
    print("Warning: Annual gross salary exceeds realistic values.")
```