Coding Challenges

Coding Bootcamp

Contents

Sequence Programming

Coding Challenge 1: Program to find the sum and average of two variables

Coding Challenge 2: Program to calculate simple interest

Coding Challenge 3: Program to calculate the discount on the total amount

Coding Challenge 4: Program to swap two numbers

Coding Challenge 5: Farmer Problem Statement

Mahesh is a farmer and owns 80 acres of land. His land is equally divided into 5 segments. He grows tomatoes in the 1st segment, potatoes in the 2nd segment, cabbage in the 3rd segment, sunflower in the 4th segment and sugarcane in the 5th segment. He is converting his land from chemical-driven farming to chemical-free farming. Mahesh starts with the conversion of vegetables into chemical-free produce. He spends the first 6 months doing the same. He then converts the sunflower land bank into chemical-free farming. This takes him another 4 months. Finally, he converts sugarcane into chemical-free farming over the next 4 months. He gets a yield of the following for tomatoes. 30% of his tomato land gives him 10 tonne yield per acre. The remaining 70% of his tomato land gives him 12 tonnes yield per acre. The selling price of tomato is Rs. 7 per Kg. The yield of potatoes is 10 tonnes per acre. He sells each kg at Rs. 20. The yield of cabbage is 14 tonnes per acre. He sells each kg at Rs. 24. The yield of sunflowers is 0.7 tonnes per acre. He sells each kg at Rs. 200. The yield of sugarcane is 45 tonnes per acre. He sells each tonne at Rs. 4,000. All the crops are sowed at the same time. Mahesh gets the above yield at the above-mentioned rate in one crop cycle across his entire land of 80 acres.
What is
a. The overall sales achieved by Mahesh from the 80 acres of land.
b. Sales realisation from chemical-free farming at the end of 11 months

Decision Making

Coding Challenge 6: Program to check if a number is even or odd

Coding Challenge 7: Program to accept name and salary. Check if their salary is >3L and display if they must pay tax

Coding Challenge 8: To find the largest of 3 numbers

Coding Challenge 9: Program to check if a year given is a leap year or not

Coding Challenge 10: 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.

Tax Calculator Problem– Hackathon

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 sCoding Challenges.
4. Apply any applicable standard deductions and rebates.

Generate reports detailing gross salary, taxable income, tax payable, and net salary.

Coding Challenge 11: 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.

Coding Challenge 12: 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.

Coding Challenge 13: Tax and Rebate Calculation

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

Tasks:

1. Calculate tax based on the following sCoding Challenges:
    - ₹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 sCoding Challenges, cess, and total tax payable.

Coding Challenge 14: 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 Salar

Coding Challenge 15: 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

Coding Challenge 16: 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.

Loops and Iterations

Coding Challenge 17: Display the Series 1,2,3,4,5,6…N

Coding Challenge 18: Display the Series 1,3,5,7,9…N

Coding Challenge 19: Display the Series 4,16,36,64…N

Coding Challenge 20: Display the Series 1,2,4,7,11,16,22…N

Coding Challenge 21: Display the Series 1,4,9,25,36,49,81…N

Coding Challenge 22: Display the Series 1,4,7,12,23…N

Coding Challenge 23: Display the Series 1,5,9,13,21,25,29,37,41…N

Coding Challenge 24: Display the Series 1,1,2,3,5,8,13,21…N

Retail Shopping– Hackathon

Retail Shopping Application with Enhanced Rules

You are tasked with developing a retail shopping application for generating itemized invoices, applying business rules for discounts, surcharges, and quantities, and providing a seamless customer experience. The application unfolds across levels, each introducing new functionality, culminating in a complete solution that includes invoice generation and a breakdown of purchases**.**

Coding Challenge 25: Basic Item Entry and Total Calculation

Objective: Allow the user to input item details (code, description, quantity, price) and calculate the total cost for the item.

Key Steps:

1. Accept item code, description, quantity, and price.
2. Compute the total for a single item.
3. Display the total for the item.

Coding Challenge 26: Iterative Item Entry and Grand Total

Objective: Enable multiple items to be added iteratively, and compute the grand total for all items.

Key Steps:

1. Use a loop to accept details for multiple items.
2. Compute the grand total by summing individual totals.
3. Display the grand total after all items have been entered.

Coding Challenge 27: Applying Discounts

Objective: Introduce business rules for modifying the grand total based on conditions.

Rules Implemented:

1. Discount: If the grand total exceeds ₹10,000, apply a 10% discount.
2. Quantity Discount: If the total quantity exceeds 20, apply an additional 5% discount on the grand total (after other discounts).

Key Steps:

1. Check conditions for discounts.
2. Adjust the grand total accordingly.
3. Display the modified total with adjustments explained.

Coding Challenge 28: Membership Discounts

Objective: Introduce a membership system where customers get an additional discount.

Rules Implemented:

1. If the customer is a member (choice: 'y'), apply an additional 2% discount on the grand total after all other adjustments.

Key Steps:

1. Prompt the user for membership status.
2. Apply the membership discount if applicable.
3. Update and display the final grand total.

Coding Challenge 29: Tax Calculation Based on Purchase Amount

Objective: Introduce tiered tax rates based on the grand total.

Rules Implemented

1. If the grand total is below ₹5,000, apply 5% tax.
2. If the grand total is between ₹5,000 and ₹20,000, apply 10% tax.
3. If the grand total exceeds ₹20,000, apply 15% tax.

Key Steps

1. Calculate the tax based on the applicable tier.
2. Add the tax to the grand total.
3. Display the tax amount and updated grand total.

Coding Challenge 30: Promotional Discounts on Specific Items

Objective: Introduce promotional discounts on specific items identified by their code.

Rules Implemented

1. If the item code matches a promotional code (e.g., PROMO10), apply a 10% discount on that item.
2. Compute the grand total considering the discounts on applicable items.

Key Steps

1. Check if the item code matches the promotional code.
2. Apply the discount to the item total.
3. Update the grand total and display the final value.

Coding Challenge 31: Payment Mode Rules

Objective: Incorporate business rules based on the selected payment method.

Rules Implemented

1. If the customer chooses Cash, no surcharge applies.
2. If the customer chooses Credit Card, apply a flat 2% surcharge on the final grand total after all adjustments.

Key Steps

1. Prompt the user to select the payment method.
2. Apply the surcharge if the method is Credit Card.
3. Display the payment method, surcharge amount, and final payable amount.

Coding Challenge 32: Minimum Purchase Requirements

Objective

Add a condition to enforce a minimum purchase value to generate an invoice.

Rules Implemented

1. If the final grand total (after discounts and taxes) is below ₹500, inform the user that the minimum purchase amount is not met.

Key Steps

1. Check the final grand total.
2. If below ₹500, display an appropriate message and terminate the process.
3. Otherwise, proceed to generate the invoice.

Coding Challenge 33: Loyalty Points

Objective: Introduce a loyalty program where customers earn points based on the final grand total.

Rules Implemented

1. For every ₹100 spent, the customer earns 1 loyalty point.
2. Display the total loyalty points earned.

Key Steps

1. Calculate loyalty points (points = grand_total // 100).
2. Display the earned points along with the invoice.

Pattern Programming and Nested Loops

Coding Challenge 34: Printing Star Pattern (N Rows)

*****

*****

*****

*****

.

.

N rows

Coding Challenge 33: Printing Number Pattern (N Rows)

11111

22222

33333

44444

.

.

N rows

Coding Challenge 34: Printing Number Pattern (N Rows)

12345

12345

12345

12345

.

.

N rows

Coding Challenge 35: Printing * Increasing Pattern (N Rows)

*
**
***
****
.
.
N rows

Coding Challenge 36: Printing number Increasing Pattern (N Rows)

1
22
333
4444
.
.
N rows

Coding Challenge 37: Printing number Increasing Pattern (N Rows)

1
12
123
1234
.
.
N rows

Coding Challenge 38: Fibonacci Series Pattern (N Rows)

1
1 2
3 5 8
13 21 34 55
.
N rows

Coding Challenge 39: Printing Pattern of Perfect Squares with Alternating Signs in N Rows

1
-4 9
-16 25 -36
49 -64 81 -100
.
N rows

Coding Challenge 40: Printing Pattern of Factorials in N Rows

1
1 2
6 24 120
.
N rows

Coding Challenge 41: Convert Number to Words Using Mathematical Logic

a. Input: 270176
b. Output: Two Seven Zero One Seven Six

Coding Challenge 42: Generate Series- 1,-5, 9,-13, 17,-21, ... N

Coding Challenge 43: Whole and Fraction value separation

Write a program to accept a double value. Separate the whole value from the fractional value and store them in two variables. Display the same.

Coding Challenge 44: Reverse of a number

Write a program to find the reverse of a number. Store the reverse value in a different variable. Display the reverse.

Working with Arrays

Coding Challenge 45: Level 0: Write a program to accept n and store the elements into the array of size n.

Coding Challenge 46: Level 1: Find the Sum of all elements in the array

Coding Challenge 47: Level 2: Find the Minimum value of all elements in the array

Coding Challenge 48: Level 3: Find the Maximum value of all elements in the array

Coding Challenge 49: Level 4: Search the given element from the array

Coding Challenge 50: Level 5: Display the number of odd and even numbers from the array

Sort and Search Arrays

Coding Challenge 51: Level 0: Write a program to accept n and store the elements into the array of size n.

Coding Challenge 52: Level 1: Reverse the given array.

Coding Challenge 53: Level 2: Sort the array in ascending or descending order based on input of user

Coding Challenge 54: Level 3: Implement Binary Search on the array.

2D Arrays

Coding Challenge 55: Write a program to create a 2D array and display its elements row-wise

Coding Challenge 56: Create a program to compute the sum of all elements in a 2D array.

Coding Challenge 57: Write a program to check if a given element exists in a 2D array

Coding Challenge 58: Write a program to store elements into a M * N matrix of integer. Display the matrix and its transpose.

Coding Challenge 59: Write a program to store elements into a M * N matrix of integer. Display the matrix and its transpose.

Coding Challenge 60: Write a program to multiply two matrices

Setting the Scene: A Day at HealWell Care Hospital

At **HealWell Care Hospital**, the day begins with the admin team preparing the system for patient care. The admin updates the list of services available for the day, ensuring accuracy and clarity. These services include a mix of diagnostics and consultations:

- General Consultation
- Blood Test
- Covid Test
- X-Ray
- CT Scan
- MRI

Once the services are set, the hospital welcomes patients who require seamless and efficient billing processes. The objective is to build a robust system where patient information, service selection, billing, and tax calculations are handled smoothly while keeping the user experience intuitive and error-free.

Consider these 2 arrays (Declare them initially)
Services: ["General Consultation", "Blood Test", "Covid Test", "X-Ray", "CT Scan", "MRI"]
Costs: [500, 300, 800, 1500, 4000, 7000]

Requirements & Levels

Level 1: Patient Walks In and Shares Their Details

Patients like **Mr. Arjun Kumar** visit the hospital. At the reception, their basic details (name, age, contact, and gender) are collected and stored in variables.
Example Input:
Name: Arjun Kumar
Age: 35
Gender: Male
Contact: 9876543210

Level 2: Displaying Services for Patient Selection

The system displays the list of services available for the day. The patient selects one or more services they wish to avail. The selected services are stored in an array (selected_services).
Example Interaction:
Available Services:
1. General Consultation
2. Blood Test
3. Covid Test
4. X-Ray
5. CT Scan
6. MRI
Patient Selected: [1, 4] // General Consultation, X-Ray
Selected Services Array: ["General Consultation", "X-Ray"]

Level 3: Fetching Costs of Selected Services

The system iterates through the selected_services array, matches the services with the main services array, and retrieves the respective costs from the costs array.

Example Output:

Selected Services: ["General Consultation", "X-Ray"]
Selected Costs: [500, 1500]

Level 4: Calculating the Total Cost

The total cost is calculated by summing up the costs of all selected services.

Example Output:

Total Cost (Before Tax): ₹2000

Level 5: Applying GST to the Bill

An 18% GST is applied to the total cost. The system calculates and adds this to the total bill amount.

Example Output:

GST (18%): ₹360
Grand Total: ₹2360

Level 6: Generating and Displaying the Invoice

The system generates a detailed invoice for the patient, including their details, services availed, individual costs, and the grand total.

Example Output:

```
------------------------------------------------
HealWell Care Hospital
Patient Invoice
------------------------------------------------

Patient Information:
Name: Arjun Kumar
Age: 35
Gender: Male
Contact: 9876543210

Services Availed:
1. General Consultation: ₹500
2. X-Ray: ₹1500

Subtotal: ₹2000
GST (18%): ₹360
Grand Total: ₹2360

Thank you for choosing HealWell Care Hospital!
------------------------------------------------
```

Level 7: Setting Up the Services of the Day (Admin Task)

The admin prepares the hospital system by entering the available services into the system. Hospital wants the flexibility to configure the services. This list is stored in an array. Each service has a name, and its corresponding cost is stored in a separate array for flexibility.

- **Array 1**: Stores the names of services (services array).
- **Array 2**: Stores the costs of the services (costs array).

Example Input:
Services: ["General Consultation", "Blood Test", "Covid Test", "X-Ray", "CT Scan", "MRI"]
Costs: [500, 300, 800, 1500, 4000, 7000]
Objective: Ensure services and costs are paired for seamless retrieval.

Level 8: Providing Discounts (Optional Enhancements)

Context
HealWell Care Hospital values its patients and strives to provide the best care at affordable rates. To improve accessibility and express gratitude, the hospital offers discounts in specific scenarios.
Discount Rules

1. **Senior Citizen Discount**:
   Patients aged 60 or above are eligible for a **10% discount** on the subtotal (before GST).
2. **High-Bill Discount**:
   If the subtotal exceeds ₹5000, an additional **5% discount** is applied on the subtotal (after any other discounts).

Steps

1. Check the patient's age to determine if the senior citizen discount applies.
2. Check the subtotal to determine if the high-bill discount applies.
3. Calculate the final total after applying discounts and GST.