

Programming Fundamentals FALL - 2024

ASSIGNMENT – 2 – SE

Deadline: Friday Oct 25, 2024

1. Make sure that you read and understand each and every instruction. If you have any questions or comments you are encouraged to discuss your problems with your colleagues (and instructors) on google classroom.
2. If there is a syntax error in the code, zero marks will be awarded in that part of the assignment.
3. Keep a backup of your work always that will be helpful in preventing any mishap and avoid last hour submissions
4. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.
5. Please ensure that only concepts covered in class are used for the assignment. For each question in your assignment, make a separate .cpp file e.g., for question 1, make q1.cpp and so on. Each file that you submit must contain your name, student-id, and assignment on top of the file in the comments.
7. Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.).
8. Rename the folder as ROLL-NUM_PROGRAM_SECTION (e.g., i240001_SE_A) and compress the folder as a zip file. (e.g., i240001_SE_A.zip). Strictly follow this naming convention, otherwise marks will be deducted.
9. Submit the .zip file on Google Classroom within the deadline.
10. The student is solely responsible for checking the final file for issues like corrupt file, viruses in the file, or mistakenly exe sent. If we cannot download the file from Google Classroom, it will lead to zero marks in the assignment.

Note: Start early so that you can finish it on time.

Question 1:

An egg distribution company uses different sizes of packings for eggs, that is, 30 eggs packing, 24 eggs packing, 18 eggs packing, 12 eggs packing and 6 eggs packing. Write a program which prompts the user to enter the total number of eggs (input validation is always must) to be packed and then calculate how many packings of each size will be possible. Also tell if there will be any eggs left to be packed.

Sample Output:

Enter total number of eggs: 100

Number of 30 eggs packing: 3 Number of leftover eggs: 10

Number of 24 eggs packing: 4 Number of leftover eggs: 4

Number of 18 eggs packing: 5 Number of leftover eggs: 10

Number of 12 eggs packing: 8 Number of leftover eggs: 4

Number of 6 eggs packing: 16 Number of leftover eggs: 4

Question 2

Write a C++ program that generates a **math exercise** for the user. The program should first ask for which operation an exercise should be generated. The Possibilities are **addition (+)**, **subtraction (-)**, **multiplication (*)**, and **division (/)**.

After selecting an operation, the user should answer the following question:

What is the maximum value for the input values of the exercise?

The answer to this question must be stored in a variable `maximumValue`. Next, the program should ask.

Are negative values allowed in the exercise?

The possible answers for the question are "Y" and "N "

The program will then generate two random numbers. If the user allows negative values, the random numbers must be generated in the range from - **maximumValue** to + **maximumValue**. If the user allows only positive values, the range is restricted from **0** to **maximumValue**. Moreover, if only positive numbers are allowed and subtraction was chosen, the second random value must be less than the first random value.

Finally, the program will show the generated exercise, depending on the chosen operation and the numbers randomly generated, and ask the user for the correct solution. If the user solves the exercise correctly, the program will print out a congratulatory message.

Question 3

Write a program in C++ to print the following pattern. Make sure your code should be generic and print the following pattern using **FOR LOOP**.

Input of 9 will print the following one.



Question 4

Create a simplified C++ program for a two-player Tic Tac Toe game using **Loops**. The game's functionality includes:

1. Displaying a 3x3 game board with a numbered grid to help players select their moves.
2. Allowing two players to take turns, one using 'X' and the other 'O.'
3. Verifying the validity of each move (i.e., the chosen cell is within the board boundaries).
4. Checking for a win condition or a draw after each move.
5. Prompting players to make their moves by specifying the 1-9.
6. Displaying the updated game board after each move.
7. Announcing the winner or a draw when the game ends.
8. Ending the game automatically after nine moves if no winner is determined.
9. Providing a player-friendly experience with clear prompts and messages.

Your task is to code and add the necessary logic to determine the game's outcome (win, draw, or continue) after each move. Also, it should also print the board after every move. Ensure that the game runs smoothly and accurately. The game should continue until a player wins or the game ends in a draw.

Note: Arrays or Functions are not allowed

Question 5

Create a program to draw a bar chart of 5 values entered by the user (≥ 0), where the y-axis is along the rows and the x-axis is along the columns of the screen.

Examples:

1) Input:

N1>=>8, N2=>6, N3=>2, N4=>10, N5=>9

```
10          *
09          * *
08 *        * *
07 *        * *
06 * *      * *
05 * *      * *
04 * *      * *
03 * *      * *
02 * * *    * *
01 * * *    * *
    1  2  3  4  5
```

Input:

N1>=>8 , N2=>6 , N3=>0 , N4=>11 , N5=>7

```
11          *
10          *
09          *
08 *        *
07 *        * *
06 * *      * *
05 * *      * *
04 * *      * *
03 * *      * *
02 * *      * *
01 * *      * *
    1  2  3  4  5
```

Input:

N1=>4 , N2=>3 , N3=>1 , N4=>0 , N5=>0

04	*				
03	*	*			
02	*	*			
01	*	*	*		
	1	2	3	4	5

Question 6

TechNova Solutions, a global leader in technology services and innovation, has introduced a performance-based bonus system to reward employees for their contributions. This system considers multiple factors, including years of service, performance rating, overtime hours, department, and work location. Emma, the HR Manager at TechNova Solutions, has reached out to your team to develop a C++ program that automates the calculation of performance bonuses for employees.

The goal is to ensure that this comprehensive system not only rewards employees fairly but also motivates them to continue excelling in their roles. Your task is to implement a C++ program that automates the calculation of performance bonuses based on the company's detailed guidelines.

Company Bonus Guidelines:

1. Years of Service:

- Employees must have worked for at least **3 years** to qualify for a bonus.
- Employees with more than **5 years of service** receive a higher percentage of their salary as a bonus.
- Employees with more than **10 years of service** receive the highest percentage bonus.

2. Performance Ratings (1 to 5): Employees are rated on a scale of 1 to 5:

- **5:** Outstanding
- **4:** Excellent
- **3:** Satisfactory
- **Below 3:** No bonus. Higher performance ratings result in a larger base bonus, with special bonuses for employees with a rating of 5

3. Bonus Structure Based on Years of Service:

○ 3 to 5 years of service:

- Performance rating of 5: **10%** of annual salary.
- Performance rating of 4: **7%** of annual salary.
- Performance rating of 3: **5%** of annual salary.
- Performance rating below 3: **No bonus**

○ More than 5 years of service:

- Performance rating of 5: **15%** of annual salary.
- Performance rating of 4: **10%** of annual salary.

- Performance rating of 3: **8%** of annual salary.

- **More than 10 years of service:**

- Performance rating of 5: **20%** of annual salary.

- Performance rating of 4: **15%** of annual salary.

- Performance rating of 3: **12%** of annual salary.

4. **Overtime Bonus:**

- Employees with more than **100 overtime hours** receive an additional **2%** of their salary as a bonus.

- Employees with more than **200 overtime hours** receive an additional **4%** of their salary.

- Employees with a performance rating of **5** receive an extra **1%** bonus on top of the overtime bonus.

5. **Department-Based Bonus:**

- Employees in the **Engineering department** receive an additional **5%** bonus on their annual salary.

- Employees in the **Sales department** receive an additional **3%** bonus.

- Employees in other departments do not receive any department-based bonus.

6. **Work Location Allowance:**

- Employees working in **Region A** receive an additional **5%** of their salary as a location allowance.

- Employees working in **Region B** receive an additional **3%** location allowance.

- Employees in other regions do not receive any additional allowance.

7. **High Salary Condition:**

- If an employee's annual salary exceeds **\$100,000**, they will receive an additional **5%** bonus based on the total calculated bonus (after all other bonuses have been added).

Objective:

You are tasked with writing a C++ program that:

- Accepts inputs for the employee's years of service, performance rating, salary, overtime hours, department, and work location.

- Calculates the base bonus based on the years of service and performance rating.
- Adds any relevant department-based bonuses, location allowances, and overtime bonuses.
- If the employee's salary exceeds \$100,000, apply an additional 5% bonus to the total.
- Displays the final bonus along with a breakdown of the bonus components (base bonus, overtime bonus, department-based bonus, location allowance, and high salary bonus).

Scenario 1: John's Outstanding Performance

John is a senior software engineer in TechNova Solutions' Engineering department. He has worked for the company for **8 years** and earns an annual salary of **\$120,000**. This year, he achieved a **performance rating of 5**, and he worked **210 overtime hours**. John is based in **Region A**, where employees receive a higher cost-of-living allowance.

- **Base Bonus:** As an employee with 8 years of service and a performance rating of 5, John is eligible for a **15%** bonus on his salary.
- **Overtime Bonus:** John worked more than **200 overtime hours**, making him eligible for a **4%** overtime bonus. Additionally, with a performance rating of 5, he receives an extra **1%** bonus on his overtime.
- **Department-Based Bonus:** As an Engineering department employee, John receives an additional **5%** bonus.
- **Location Allowance:** Working in **Region A**, John qualifies for an extra **5%** of his salary.
- **High Salary Bonus:** Since John's salary exceeds **\$100,000**, he is eligible for an extra **5%** bonus on the total calculated bonus.

Scenario 2: Sarah's Sales Success

Sarah has been a **sales executive** at TechNova Solutions for **4 years**. She has earned a **performance rating of 4** this year, worked **110 overtime hours**, and earns an annual salary of **\$85,000**. She is based in **Region B**, where employees receive a moderate location allowance.

- **Base Bonus:** Sarah is eligible for a **7%** bonus for her 4 years of service and a performance rating of 4.
- **Overtime Bonus:** With more than **100 overtime hours**, she qualifies for a **2%** bonus.
- **Department-Based Bonus:** Sarah, working in the **Sales department**, qualifies for a

3% bonus.

- **Location Allowance:** Working in **Region B**, she receives an additional **3%** allowance.

Scenario 3: Alex's Limited Experience

Alex is a **marketing assistant** who has been with TechNova Solutions for **2 years**. He has a **performance rating of 3** and an annual salary of **\$60,000**. He has worked **90 overtime hours** this year but does not meet the minimum years of service requirement for a bonus.

- **No Bonus:** Since Alex has only worked for 2 years, he does not qualify for any performance bonus.

Program Requirements:

- **Accept the following inputs from the user:**
 - Years of service
 - Performance rating (1 to 5)
 - Annual salary
 - Overtime hours
 - Department (Engineering, Sales, Other)
 - Work location (Region A, Region B, Other)
- **Bonus Calculations:**
 - Calculate the base bonus based on the employee's years of service and performance rating.
 - Add the overtime bonus based on the number of overtime hours.
 - Include any department-based bonus (for Engineering or Sales).
 - Apply the location allowance based on the work location.
 - If the employee's salary exceeds \$100,000, apply an additional **5% bonus** to the total calculated bonus.
- Display the **final bonus** amount and provide a **breakdown** of the bonus components:
 - Base bonus
 - Overtime bonus
 - Department-based bonus
 - Location allowance

High salary bonus (if applicable)

Constraints:

- The solution must be generic and not involve directly outputting values via cout. As Direct cout will result in direct zero.
- Use Switch for Menu-Driven Program.