# 

**OCCUPATIONAL CERTIFICATE:**

**SOFTWARE ENGINEER**

**Week Four Practical (C#)**

## **Mathematics**

## **MAT621**

**2025**

**OCCUPATIONALCERTIFICATE:**

**SOFTWARE ENGINEER**

**Formative Assessment 1**

# Software Design Engineering

## **SDE631**

**2025**

**C# Probability Calculator Assignment**

Question: Probability Calculator Program (100 Marks)

Design and implement a C# console application that calculates the probability of the union of two events, supporting both mutually exclusive and non-mutually exclusive(inclusive) events. The program should meet the following requirements:

**Requirement 1**: User Interface (15 marks): Provide a clear console interface that explains the programs purpose and allows the user to choose between calculating probabilities for mutually exclusive or inclusive events.

**Requirement 2**: Input Collection (20 marks): Prompt for and collect probabilities P(A) and P(B) for both event types, and P(A and B) for inclusive events. Ensure all inputs are valid numbers between 0 and 1.

**Requirement 3**: Probability Calculations (20 marks): Correctly calculate:

• Mutually exclusive events: P(A ∪ B) = P(A) + P(B)

• Inclusive events: P(A ∪ B) = P(A) + P(B) − P(A ∩ B)

**Requirement 4:** Input Validation (15 marks): Validate that probabilities are between 0 and 1, and for inclusive events, ensure P(A and B) is not

greater than P(A) or P(B).

**Requirement 5**: Error Handling (15 marks): Handle invalid inputs (e.g., nonnumeric inputs) gracefully using exception handling, providing clear error messages.

**Requirement 6**: Program Flow (10 marks): Allow the user to perform multiple calculations in a single session and choose to exit the program.

**Requirement 7:** Code Quality (15 marks): Use clear variable names, include comments, organize code into methods for modularity, and follow C# coding conventions.

Mark Allocation

• User Interface: 15 marks

• Input Collection: 20 marks

• Probability Calculations: 20 marks

• Input Validation: 15 marks

• Error Handling: 15 marks

• Program Flow: 10 marks

• Code Quality: 15 marks

• Total: 100 marks

How the Solution Meets Requirements

• User Interface (15/15): The program starts with a clear welcome message explaining its purpose and the formulas used. It prompts the user to select the event type (1 or 2) and guides input collection.

• Input Collection (20/20): The program collects P(A) and P(B) for both event types and P(A and B) for inclusive events using clear console prompts.

• Probability Calculations (20/20): Implements correct formulas for mutually exclusive (P(A) + P(B)) and inclusive (P(A) + P(B) − P(A ∩ B)) events, with results formatted to 4 decimal places.

• Input Validation (15/15): Validates that probabilities are between 0 and 1 using

the Parse Probability method and checks that P(A and B) does not exceed P(A)or P(B) for inclusive events.

• Error Handling (15/15): Uses try-catch blocks to handle Format Exception for non-numeric inputs, ArgumentOutOfRangeException for invalid probabilities, and a general Exception for unexpected errors.

• Program Flow (10/10): Implements a while loop controlled by a boolean flag,allowing multiple calculations until the user chooses to exit by entering ’n’.

• Code Quality (15/15): Uses meaningful variable names (e.g., probA, probB), includes XML documentation for methods, separates logic into modular methods, and follows C# conventions with proper indentation and structure.

**How to Run**

Compile and run the program in a C# environment such as Visual Studio. The program will prompt for inputs, perform calculations, and display results. Invalid inputs will trigger error messages, and the user can choose to continue or exit after each calculation