# 

**OCCUPATIONAL CERTIFICATE:**

**SOFTWARE ENGINEER**

**Week Three Practical (C#)**

## **Mathematics**

## **MAT621**

**2025**

**OCCUPATIONALCERTIFICATE:**

**SOFTWARE ENGINEER**

**Formative Assessment 1**

# Software Design Engineering

## **SDE631**

**2025**

You are tasked with writing a C# console application to model the continuous exponential growth of an investment under compound interest using the formula ( A = P e^{rt} ), where:

* ( A ) is the final investment value (dependent variable),
* ( P ) is the principal (initial investment amount),
* ( r ) is the annual interest rate (as a decimal, e.g., 5% = 0.05),
* ( t ) is the time in years (independent variable),
* ( e ) is the base of the natural logarithm (approximately 2.71828).

Your program must allow the user to input the principal ( P ) and interest rate ( r ). It should vary the time ( t ) over a user-specified range, calculate the corresponding investment values, and analyze how changes in ( t ) affect ( A ). The program should provide a detailed output, including a table of results, growth analysis, and an option to save results to a text file.

**Requirements**

1. **User Input**:
   * Prompt the user to enter the principal ( P ), a positive double value).
   * Prompt the user to enter the interest rate ( r ), a double value, positive for growth or negative for depreciation).
   * Prompt the user to enter the time range (start time, end time, and step size for ( t ), all positive doubles, where start time < end time and step size is reasonable).
2. **Calculations**:
   * Use the continuous compounding formula ( A = P e^{rt} ) to compute ( A ) for each time value in the specified range.
   * Vary ( t ) from the start time to the end time, incrementing by the step size.
   * Use the Math.Exp method in C# to compute ( e^{rt} ).
3. **Output**:
   * Display a table showing time ( t ) and corresponding investment value( A ), formatted with 2 decimal places for both.
   * Provide an analysis of the growth trend, e.g., whether the investment is growing or depreciating and the percentage change from the initial value to the final value.
   * Offer an option to save the table to a text file named "InvestmentGrowth.txt".
4. **Error Handling**:
   * Validate all inputs to ensure they are valid (e.g., positive ( P ), reasonable step size, start time < end time).
   * Handle exceptions (e.g., non-numeric inputs, file I/O errors) with appropriate error messages.
5. **Code Quality**:
   * Use meaningful variable names and follow C# naming conventions.
   * Include comments to explain the logic.
   * Organize the code into methods for modularity (e.g., input, calculation, output, file handling).