XJTLU Entrepreneur College (Taicang) Cover Sheet

Module code and Title	**DTS102TC Programming with C++**
School Title	**School of Artificial Intelligence and Advanced Computing**
Assignment Title	**Coursework 1 (Individual Assessment)**
Submission Deadline	**5 pm China time (UTC+8 Beijing) on** Fri. 18th. Oct. 2024
Final Word Count	**NA**
If you agree to let the university use your work anonymously for teaching and learning purposes, please type **"yes"** here.	

I certify that I have read and understood the University's Policy for dealing with Plagiarism, Collusion and the Fabrication of Data (available on Learning Mall Online). With reference to this policy I certify that:

- My work does not contain any instances of plagiarism and/or collusion.
  My work does not contain any fabricated data.

By uploading my assignment onto Learning Mall Online, I formally declare that all of the above information is true to the best of my knowledge and belief.

Scoring – For Tutor Use	
Student ID	

Stage of Marking	**Marker Code**	**Learning Outcomes Achieved（F/P/M/D）(please modify as appropriate)**			**Final Score**
		A	**B**	**C**	
1st Marker – red pen					
Moderation – green pen	**IM Initials**	The original mark has been accepted by the moderator (please circle as appropriate):			Y / N
		Data entry and score calculation have been checked by another tutor (please circle):			Y
2nd Marker if needed – green pen					

For Academic Office Use			**Possible Academic Infringement (please tick as appropriate)**	
Date Received	**Days late**	**Late Penalty**	☐ **Category A**	Total Academic Infringement Penalty (A,B, C, D, E, Please modify where necessary) _______________
			☐ **Category B**	
			☐ **Category C**	
			☐ **Category D**	
			☐ **Category E**	

DTS102TC Programming with C++
Coursework 1 (Individual Project Assessment)

Deadline: 5:00 pm China time (UTC+8 Beijing) on Friday 18th. Oct. 2024

Percentage in final score: 50%

Maximum score: 100 marks (100% individual marks)

Learning outcomes assessed:

A. Demonstrate knowledge and understanding of basic principles of C++ programming language.

B. Demonstrate knowledge and understanding of basic software development process.

Late policy: 5% of the total marks available for the assessment shall be deducted from the assessment mark for each working day after the submission date, up to a maximum of five working days.

Notice:

- Please read the coursework instructions and requirements carefully. Not following these instructions and requirements may result in loss of marks.
- The assignment must be typed in an MS Word and converted to a PDF document. The document must be submitted via Learning Mall Online to the Gradescope. Only electronic submission is accepted and no hard copy submission. **All submissions should be written in English.**
- All students must download their file and check that it is viewable after submission. Documents may become corrupted during the uploading process (e.g. due to slow internet connections). However, students themselves are responsible for submitting a functional and correct file for assessments.
- Please download the ***Source Code Template*** from Learning Mall Online. **Do not change the file name of each code script.**
- Academic Integrity Policy is strictly followed.

Overview

The purpose of this task is to gain experience in C++ programming and software development skills. You are expected to write a C++ program to solve each question.

For each question, you need to write the code to display the results which is the same as the sample run. Submit your code in Gradescope to test the program implementation. Also, you will write a short description for each question in the report to analyse your methods and discuss the results with your test cases. The code quality such as naming rules of variables and comments of functions will also be evaluated.

Question 1. *Financial application: future investment value* (10 marks)

Write a program that reads in investment amount, annual interest rate, and number of years, and displays the future investment value using the following formula:

futureInvestmentValue = investmentAmount x (1 + monthlyInterestRate)^ (numberOfYears x 12)

For example, if you enter amount 1000.56, annual interest rate 4.25%, and number of years 1, the future investment value is $1043.92.

Sample Run

```
Enter investment amount: 1000.56
Enter annual interest rate in percentage: 4.25
Enter number of years: 1
Accumulated value is $1043.92
```

Question 2. *Science: day of the week* (10 marks)

Zeller's congruence is an algorithm developed by Christian Zeller to calculate the day of the week.

The formula is:

$$h = \left(q + \frac{26(m+1)}{10} + k + \frac{k}{4} + \frac{j}{4} + 5j \right) \% \, 7$$

where

- h is the day of the week (0: Saturday, 1: Sunday, 2: Monday, 3: Tuesday, 4: Wednesday, 5: Thursday, 6: Friday).
- q is the day of the month.
- m is the month (3: March, 4: April, . . . , 12: December). January and February are counted as months 13 and 14 of the previous year.
- j is year/100
- k is the year of the century (i.e., year % 100)

Note that all divisions in this exercise perform an integer division.

Write a program that enters a year, month, and day of the month, and displays the name of the day of the week.

Sample Run 1

```
Enter year (e.g., 2012): 2015
Enter month (1-12): 1
Enter the day of the month (1-31): 25
Day of the week is Sunday
```

Sample Run 2

```
Enter year (e.g., 2012): 2012
Enter month (1-12): 5
Enter the day of the month (1-31): 12
Day of the week is Saturday
```

(Hint: January and February are counted as 13 and 14 in the formula, so you need to convert the user input 1 to 13 and 2 to 14 for the month and change the year to the previous year.)

Question 3. *Order three cities* (10 marks)

Write a program that enters three cities and displays them in ascending order. City names may contain spaces.

Sample Run

```
Enter the first city: Shanghai
Enter the second city: Suzhou
Enter the third city: Beijing
The three cities in alphabetical order are Beijing Shanghai Suzhou
```

Question 4. *Check password* (10 marks)

Some websites impose certain rules for passwords. Suppose the password rules are as follows:

- A password must have at least eight characters.
- A password must consist of only letters and digits.
- A password must contain at least two digits.

Write a program that enters a password and displays valid password if the rules are followed or invalid password otherwise.

Sample Run

```
Enter a string for password: DTS102TC
valid password!
Enter a string for password: C++ Programming
invalid password!
```

Question 5. *Algebra: solve 2 × 2 linear equations* (15 marks)

You can use Cramer's rule to solve the following 2 × 2 system of linear equation:

$$\begin{array}{l} ax + by = e \\ cx + dy = f \end{array} \qquad x = \frac{ed - bf}{ad - bc} \qquad y = \frac{af - ec}{ad - bc}$$

Write a function with the following header:

```
void solveEquation(double a, double b, double c, double d, double e, double
f, double& x, double& y, bool& isSolvable)
```

If *ad* - *bc* is 0, the equation has no solution and `isSolvable` should be `false`.

Write a program that enters *a, b, c, d, e,* and *f* and displays the result.

If *ad* - *bc* is 0, report that "The equation has no solution."

Sample Run

```
Enter a, b, c, d, e, f: 9.0 4.0 3.0 -5.0 -6.0 -21.0
x is -2.0 and y is 3.0
Enter a, b, c, d, e, f: 1.0 2.0 2.0 4.0 4.0 5.0
The equation has no solution
```

Question 6. *Financial application: compute the future investment value* (15 marks)

(Financial application: compute the future investment value)

Write a function that computes future investment value at a given interest rate for a specified number of years. The future investment is determined using the formula in Question 1.

Use the following function header:

```
double futureInvestmentValue(double investmentAmount, double
monthlyInterestRate, int years)
```

For example,

`futureInvestmentValue(10000, 0.05/12, 5)` returns 12833.59.

Write a test program that prompts the user to enter the investment amount (e.g., 1000) and the interest rate (e.g., 9%) and prints a table that displays future value for the years from 1 to 30, as shown below:

Sample Run

```
The amount invested: 1000
Annual interest rate: 9
Years Future Value
1     1093.81
2     1196.41
...
29    13467.25
30    14730.58
```

Question 7. *Statistics: compute mean and standard deviation* (15 marks)

Apply the below formula to compute the standard deviation of n numbers.

$$mean = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n} \qquad deviation = \sqrt{\frac{\sum_{i=1}^{n} (x_i - mean)^2}{n - 1}}$$

To compute deviation with this formula, you have to store the individual numbers using an array, so that they can be used after the mean is obtained.

Your program should contain the following functions:

```
// Compute the mean of an array of double values
double mean(const double x[], int size)

// Compute the deviation of double values
double deviation(const double x[], int size)
```

Write a test program that enters 10 numbers and displays the mean and deviation, as shown in the following sample run:

Sample Run

```
Enter ten numbers: 1.9 2.5 3.7 2 1 6 3 4 5 2
The mean is 3.11
The standard deviation is 1.55738
```

Question 8. *Markov matrix* (15 marks)

An n by n matrix is called a positive Markov matrix if each element is positive and the sum of the elements in each column is 1. Write the following function to check whether a matrix is a Markov matrix.

```
const int SIZE = 3;
bool isMarkovMatrix(const double m[][SIZE]);
```

Write a test program that enters a 3 by 3 matrix of double values and tests whether it is a Markov matrix. Here are sample runs:

Sample Run 1

```
Enter a 3-by-3 matrix row by row:
0.15 0.875 0.375
0.55 0.005 0.225
0.30 0.12 0.4
It is a Markov matrix
```

Sample Run 2

```
Enter a 3-by-3 matrix row by row:
0.95 -0.875 0.375
0.65 0.005 0.225
0.30 0.22 -0.4
It is not a Markov matrix
```

Submission

Each individual student must submit the following files:

- **Report:** A ***Student_ID.pdf*** file contains a cover letter with your information. This is a short report involves the program design, test results, and analysis comments for each question. **The report should not exceed 10 pages.**
- **Code:** A ***Student_ID.zip*** file should include your program implementation, with all source code files, i.e., Question1.cpp, Question2.cpp, Question3.cpp, etc. You must submit this source code on **Gradescope**, including all questions, even if you have not answered some of them

End of Coursework

Appendix: Individual Assessment Marking Criteria

Tasks	100	Marking Criteria	Marks
Question 1	10	**Reports [5 marks]** • Uses the correct formula for calculating future investment value [2 mark] • Displays the calculated future investment value in a clear and readable format, including the dollar sign and two decimal places [2 mark] • Clarity comments and conciseness in the code [1 mark] **Program execution by Gradescope [5 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [5 marks]	
Question 2	10	**Reports [5 marks]** • Implements Zeller's congruence algorithm correctly, including all calculations and adjustments [2 marks] • Defines a function that takes year, month, and day as arguments and returns the day of the week as a string [2 marks] • Uses clear and concise code with meaningful variable names and comments [1 mark] **Program execution by Gradescope [5 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [5 marks]	
Question 3	10	**Reports [5 marks]** • Uses an appropriate sorting algorithm to sort the city names in ascending order, regardless of case [2 marks] • Displays the ordered cities in a clear and readable format, including the correct order and capitalization [2 marks] • Uses clear and concise code with meaningful variable names and comments [1 mark] **Program execution by Gradescope [5 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [5 marks]	

Question 4	10	**Reports [5 marks]** • Implements the password validation rules correctly, including checking for length, alphanumeric characters, and minimum digit count [2 marks] • Defines a function that takes a password as an argument and returns True if the password is valid, False otherwise [2 marks] • Uses clear and concise code with meaningful variable names and comments [1 mark] **Program execution by Gradescope [5 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [5 marks]	
Question 5	15	**Reports [7 marks]** • Implements Cramer's rule correctly to solve the 2x2 linear equation system. [2 mark] • Defines a function solve_equation() with appropriate arguments (coefficients, variables, and a flag for solvability) and return type (void). [2 mark] • Displays the calculated values of x and y if the equation has a solution, or displays a message indicating that the equation has no solution. [2 mark] • Uses clear and concise code with meaningful variable names and comments [1 mark] **Program execution by Gradescope [8 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [8 marks]	
Question 6	15	**Reports [7 marks]** • Implements the future investment value formula correctly, including calculations and adjustments for monthly interest rate and number of years. [2 mark] • Defines a function futureInvestmentValue() with appropriate arguments (investment amount, monthly interest rate, and years) and a meaningful return type (future value). [2 mark] • Displays a table with the years and corresponding future investment values,	

		formatted with appropriate headers and decimal places. [2 marks] • Uses clear and concise code with meaningful variable names and comments [1 mark] **Program execution by Gradescope [8 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [8 marks]	
Question 7	15	**Reports [9 marks]** • Implements the formulas for mean and standard deviation correctly, including calculations and adjustments for sample deviation. **[2 marks]** • Defines two functions: mean() for calculating the mean and deviation() for calculating the standard deviation, both with appropriate arguments (array of values and size) and meaningful return types. **[4 marks]** • Displays the calculated mean and standard deviation with appropriate labels and formatting **[2 marks]** • Uses clear and concise code with meaningful variable names and comments **[1 mark]** **Program execution by Gradescope [6 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [6 marks]	
Question 8	15	**Reports [9 marks]** • Implements the definition of a Markov matrix correctly, including checking for positive elements and column sums equal to 1. [2 marks] • Defines a function isMarkovMatrix() with an appropriate argument (matrix) and a meaningful return type (boolean). [4 marks] • Displays a clear message indicating whether the matrix is a Markov matrix or not. [2 marks] • Uses clear and concise code with meaningful variable names and comments [1 mark] **Program execution by Gradescope [6 marks]** • Program testing and execution. Performs the calculation correctly and arrives at the correct answer for the given example [6 marks]	