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 (Assignment)**
Submission Deadline	**5 pm China time (UTC+8 Beijing) on** Fri. 28th. Nov. 2025
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 (Assignment)

Deadline: 5:00 pm China time (UTC+8 Beijing) on Friday 28th. Nov. 2025

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 objective of this assignment is to develop proficiency in C++ programming and software development skills. You are required to write a C++ program to address each question.

For each question, you must write code to generate the required results and submit it via Gradescope for evaluation of program implementation. Additionally, for each question in your report, you will need to prepare a brief report analyzing your methods and discussing the results in conjunction with your test cases. Code quality—including variable naming conventions and function comments—will also be assessed.

Section A. Fundamental of C++ Programming (70marks)

In this section, there are 9 questions. Please use the code template available on the Learning Mall and submit your code solutions via Gradescope to pass the test cases. **Note that you are limited to 5 submissions per question on Gradescope; exceeding this maximum will result in a loss of full marks for that question.**

Question 1. *Algebra: solve quadratic equations* (5 marks)

The two roots of a quadratic equation $ax^2 + bx + x = 0$ can be obtained using the following formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$b^2 - 4ac$ is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots.

Follow the code template to write a program that has values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is 0, display one root. Otherwise, display "The equation has no real roots."

Note that you can use pow(x, 0.5) to compute $\sqrt{x}$. Here are some sample examples.

Sample Run

```
Enter a, b, c: 1.0 -3 2
The roots are: 2.0 and 1.0
Enter a, b, c: 1 -2 1
The root is: 1.0
Enter a, b, c: 1 0 1
The equation has no real roots
```

Question 2. *Geometry: area of a regular polygon* (5 marks)

A regular polygon is an n-sided polygon in which all sides are of the same length and all angles have the same degree (i.e., the polygon is both equilateral and equiangular).

The formula for computing the area of a regular polygon is:

$$Area = \frac{n \times s^2}{4 \times \tan\left(\frac{\pi}{n}\right)}$$

where ***s*** is the length of a side. Follow the code template to write a program that has the number of sides and their length of a regular polygon and displays its area.

Sample Run

```
Enter the number of sides: 5
Enter the length of a side: 6.5
The area of the polygon is 72.69
```

Question 3. ***Count positive and negative numbers and compute the average of numbers*** **(5 marks)**

Follow the code template to write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Display the average as a floating-point number.

Sample Run

```
Enter an integer value, the input ends if it is 0: 1 2 -1 4 0
The number of positives is 3
The number of negatives is 1
The total value is 6
The average value is 1.50
```

Question 4. ***Binary to decimal*** **(5 marks)**

Follow the code template to write a function that returns a decimal number from a binary string.

For example, *binaryString* 10001 is 17 (1*2^4+0*2^3+0*2^2+0*2+1=17).

So, the function *solve_bin_to_dec*("10001") returns 17.

Question 5. ***Print distinct numbers*** **(5 marks)**

Follow the code template to write a program that reads in 10 numbers and displays distinct numbers (i.e., if a number appears multiple times, it is displayed only once). The numbers are displayed in the order of their input and separated by exactly one space.

(Hint: Read a number and store it to an array if it is new. If the number is already in the array, discard it. After the input, the array contains the distinct numbers.)

Sample Run

```
Enter ten numbers: 1 2 3 2 1 6 3 4 5 2
The number of distinct numbers is 6
The distinct numbers are: 1 2 3 6 4 5
```

Question 6. *Sum elements in each column* (10 marks)

Write a function that returns the sum of all the elements in a specified column in a matrix using the following header:

```
double sumColumn(const double m[][4], int rowSize, int columnIndex)
```

Follow the code template to write a program that reads a matrix and displays the sum of each column.

Sample Run

```
Enter a 3-by-4 matrix row by row:
1.5 2 3 4
5.5 6 7 8
9.5 1 3 1
Sum of the elements at column 0 is 16.5
Sum of the elements at column 1 is 9
Sum of the elements at column 2 is 13
Sum of the elements at column 3 is 13
```

Question 7. *The Rectangle class* (10 marks)

Design a class named *Rectangle* to represent a rectangle. The class contains:

- Two double data fields named *width* and *height* that specify the width and height of the rectangle.
- A no-arg constructor that creates a rectangle with width 1 and height 1.
- A constructor that creates a rectangle with the specified width and height.
- The accessor and mutator functions for all the data fields.
- A function named **getArea()** that returns the area of this rectangle.
- A function named **getPerimeter()** that returns the perimeter.

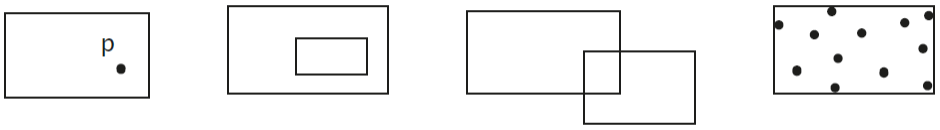
Implement the class. Write a test program that creates two Rectangle objects. Assign width 4 and

height 40 to the first object and width 3.5 and height 35.9 to the second. In the report, display the width, height, area, and perimeters of the first object and then the second object.

Question 8. ***Geometry: The Rectangle2D class*** **(15 marks)**

Define the *Rectangle2D* class that contains:

- Two double data fields named x and y that specify the center of the rectangle with constant get functions and set functions. (Assume that the rectangle sides are parallel to x- or y-axes.)
- The double data fields width and height with constant get functions and set functions.
- A no-arg constructor that creates a default rectangle with (0, 0) for (x, y) and 1 for both width and height.
- A constructor that creates a rectangle with the specified x, y, width, and height.
- A constant function getArea() that returns the area of the rectangle.
- A constant function getPerimeter() that returns the perimeter of the rectangle.
- A constant function contains(double x, double y) that returns true if the specified point (x, y) is inside this rectangle. See Figure (a).
- A constant function contains(const Rectangle2D &r) that returns true if the specified rectangle is inside this rectangle. See Figure (b).
- A constant function overlaps(const Rectangle2D &r) that returns true if the specified rectangle overlaps with this rectangle. See Figure (c).



(a) (b) (c) (d)

In the report, draw the UML diagram for the class. Implement the class with the testing cases.

Question 9. ***Geometry: find the bounding rectangle*** **(10 marks)**

A bounding rectangle is the minimum rectangle that encloses a set of points in a two-dimensional plane, as shown in Figure (d). Write a function that returns a bounding rectangle for a set of points in a two-dimensional plane, as follows:

```
const int SIZE = 2;

Rectangle2D getRectangle(const double points[][SIZE]);
```

Write another function that returns a pointer to the bounding rectangle as follows:

```
Rectangle2D* getRectanglePointer(const double points[][SIZE]);
```

The *Rectangle2D* class is defined in Question 8.

Follow the code template to write a program that enters five points and displays the bounding rectangle's center, width, and height.

Sample Run

```
Enter five points: 1.0 2.5 3 4 5 6 7 8 9 10
The bounding rectangle's center (5.0, 6.25), width 8.0, height 7.5
The bounding rectangle's center (5.0, 6.25), width 8.0, height 7.5
```

Section B. Assignment Report (30 marks)

1) For all questions, correctly describe and analyse the key functions. Provide the program testing and execution which performs the calculation and arrives at the correct answer for the test cases **(2 marks * 9 = 18 marks)**
2) For all questions, provide clear and concise code with meaningful variable names and comments **(1 marks * 9 = 9 marks)**
3) Report quality: logical structure follows the template, adherence to 5-page limit, PDF format, no major grammar/spelling errors, etc. **(3 marks)**

Submission

Each individual student must submit the following files:

- **Report:** A ***Student Name_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 5 pages.**
- **Code:** A ***Student Name_ID.zip*** file should include your program implementation, with all source code files. Please do not change the file names from the source code template. You must submit this source code on **Gradescope**, including all questions, even if you have not answered some of them.
- You are allowed **5 resubmission attempts** for your code on Gradescope. If you exceed this limit, your score will be based on your last submission.

End of Coursework