

## Roadmap to Finish the Assignment and Get Full Marks

### STEP 1: Understand the Requirements

Read and understand:

- You are building a console app in C++ using Object-Oriented Programming (OOP)
- You'll use inheritance, UML, testing, and good code style
- You must create 6 classes: Coordinates, Shape, Rectangle, Square, Circle, Triangle plus ShapeList and ShapeManagement.

### STEP 2: Set Up Your Project Folder

Create a folder with:

/project/

main.cpp

Coordinates.h / Coordinates.cpp

Shape.h / Shape.cpp

Rectangle.h / Rectangle.cpp

Square.h / Square.cpp

Circle.h / Circle.cpp

Triangle.h / Triangle.cpp

ShapeList.h / ShapeList.cpp

ShapeManagement.h / ShapeManagement.cpp

test\_output.txt

UML-Diagram.png (or .drawio)

Report.docx (your written report)

Initialize Git (optional but recommended)

### STEP 3: Implement Classes One by One

Start from the simplest to the most complex:

### 3.1 Coordinates class

- Methods: getX(), getY(), distance(), translate(), scale(), display()

### 3.2 Abstract Shape class

- Attribute: position (a Coordinates object), sides
- Virtual Methods: getArea(), getPerimeter(), display(), translate(), scale()

### 3.3 Rectangle class (inherits Shape)

- Attributes: width, length
- Override methods: getArea(), getPerimeter(), scale(), display()

### 3.4 Square class (inherits Shape)

- Attribute: side
- Override similar to rectangle

### 3.5 Circle class (inherits Shape)

- Attribute: radius
- Override  $\text{getArea} = \pi r^2$ ,  $\text{getPerimeter} = 2 \pi r$

### 3.6 Triangle class (inherits Shape)

- Attributes: 3 Coordinates objects (a, b, c)
- Override: getArea() using Herons Formula, getPerimeter = a+b+c, translate() = move all 3 vertices, scale() = custom logic to scale all points from centroid, display()

## **STEP 4: Implement ShapeList class**

- Internal list of Shape\* (polymorphism)
- Methods: addShape(), removeShape(pos), getShape(pos), area(pos), perimeter(pos),

translateShapes(dx, dy), scale(factor, sign), display()

### **STEP 5: Implement ShapeManagement class (main menu)**

Menu with options:

1. Add a shape (ask user what type and its dimensions)
2. Remove shape by position
3. Display one shape
4. Show area & perimeter of shape
5. Display all shapes
6. Translate all shapes
7. Scale all shapes
0. Quit

### **STEP 6: Build the UML Diagrams**

Draw using pen/paper or software like draw.io

You need:

- UML Class diagram (with attributes + methods)
- UML Relationship diagram (with inheritance arrows)

### **STEP 7: Testing**

Use the table in the ATI:

Action	Expected Result	Actual Result
--------	-----------------	---------------

Fill it for each operation (create shape, display, remove, translate, scale). Paste this table in your report.

### **STEP 8: Write the Report**

- Arial 11, single spacing, 5001000 words
- Sections: Introduction, UML Design, Implementation (key decisions, OOP features used), Testing (use the table), Problems Encountered (if any), Conclusion

### **STEP 9: Final Checks**

- Use meaningful variable names (camelCase)
- Add comments to every method
- Spellcheck your report
- Format properly (font, headings, page numbers)

## **STEP 10: Submission**

Submit two things:

1. .zip file of your project folder (all code files)
2. Report.docx or .pdf including UML, explanation, test table, and code in appendix

### **Want Full Marks? Focus On:**

- Clean code structure with good class design
- Clear UML diagram
- Working features in menu
- Fully filled test table with screenshots/output
- Well-written report with structure and formatting