



EBU4201 Introductory Java Programming 2024/25

Mini-Project

Introduction

You are expected to work collaboratively in your assigned group for this Mini-Project, as per the list of "Mini-Project Groups" published on QMPlus. Begin by carefully reading the project specification together, then agree on how tasks will be allocated and establish a clear timeline for completion.

Submission on QMPlus must be done by ONE of the group members by the deadline indicated under the "Mini-Project Submission" task. See page 12 for details of the submission requirements.

Assessment will consist of two parts, i.e. group submission made on QMPlus and individual student performance during the mini-viva. Details of marking criteria will be provided separately.

Feedback forms will be provided with a breakdown of marks and comments.

Specification

A primary school in London has reached out to your team to develop an interactive Java application. The purpose of this tool is to support their children in grasping the fundamental concepts of shapes, angles, and other geometric topics, which many of them are currently struggling to understand. Below are the user requirements they have provided.

We are QMUL London School, home to over 500 children aged between 5 and 10 years, enrolled in Key Stage 1 (Years 1–2) and Key Stage 2 (Years 3–4). Due to the large class sizes, it becomes challenging to provide individualised attention. Therefore, we are seeking your assistance in designing a user-friendly Graphical User Interface (GUI) based application called "Shapeville" that can act as a teaching assistant, helping children learn and practise basic geometry.

Inclusivity and accessibility: our population includes 14 children who are colourblind, so it is essential that the design is inclusive and accessible to support their needs. In addition, some children have short attention spans, therefore the application should be able to maintain engagement and motivation.

Following a thorough requirement analysis, a list of core features has been identified which must be included at a minimum in the final application as listed below.

Shapeville Introduction

Once launched, the initial interface of the application should display:

- Information about the available game levels.
- A “progress bar” to help users track their advancement during the session.
- A button labelled “End session” - when the user clicks on this button, an appropriate message with the current score/progress must be displayed before gracefully terminating the application. E.g. “You have achieved 21 points in this session. Goodbye!”.

A button labelled “Home” should facilitate the user to return to the above screen at any time.

Key Stage (KS)1:

Children will engage in two core tasks:

- Task 1: Identification of Shapes
- Task 2: Identification of Angle Types

Key Stage (KS) 2:

Children will work through two core tasks, along with two bonus challenges:

- Task 3: Area Calculation of shapes
- Task 4: Area and Circumference Calculation of Circle

Bonus Tasks:

- Bonus 1: Compound Shape Area Calculation
- Bonus 2: Sector Area Calculation

Scoring and Feedback:

Two levels of scoring, Basic and Advanced, are used depending on the task. Table 1 show the number of points awarded in each level, depending on the number of attempts taken to provide a correct answer.

Level Attempt	Basic	Advanced
One	3	6
Two	2	4
Three	1	2

Table 1 Points awarded based on levels and attempts

An appropriate message, e.g. “Great job!” must also be displayed every time points are awarded.

Task 1: Identification of Shapes

This task consists of two sub-tasks:

- **2D Shapes** *[Basic level scoring]*
- **3D Shapes** *[Advanced level scoring]*

Figure 1 and **Figure 2** provide a list of essential 2D and 3D shapes respectively.

For each sub-task (i.e. 2D and 3D), the application should function as follows:

- Display one shape at a time.
- Prompt the user to enter the name of the shape.
- If all 3 attempts are incorrect, the application should display the correct name of the shape.
- Process continues until the user has **practised** all 4 shapes OR clicks on the “Home” button.

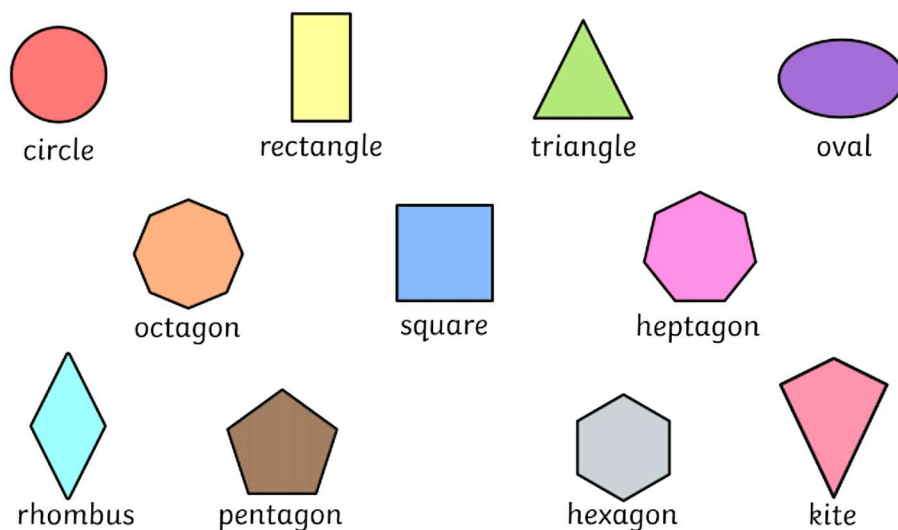


Figure 1: Basic 2D Shapes for Identification Task

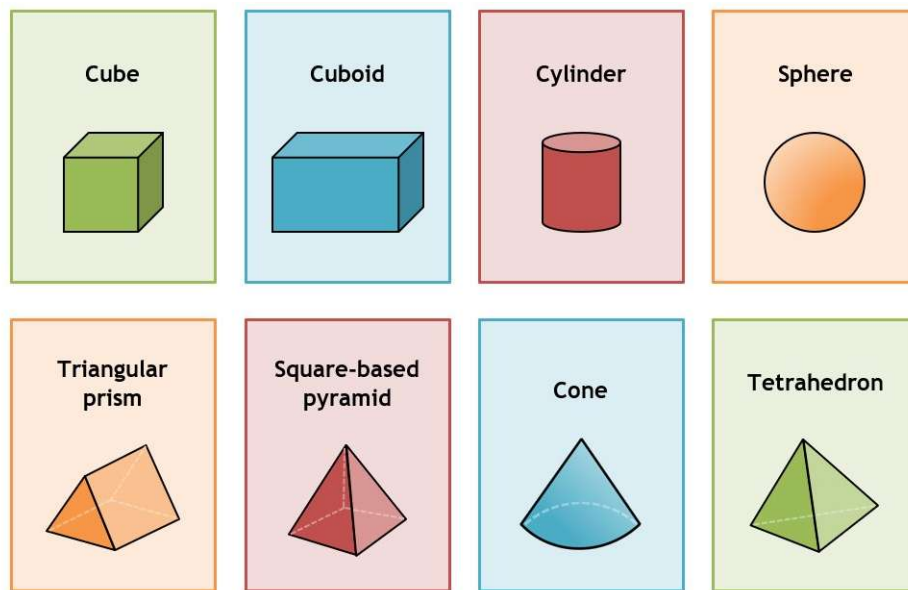


Figure 2: Basic 3D Shapes for Identification Task

Task 2: Identification of Angle Types

[Basic level scoring]

Figure 3 shows the key angle types the users must learn.

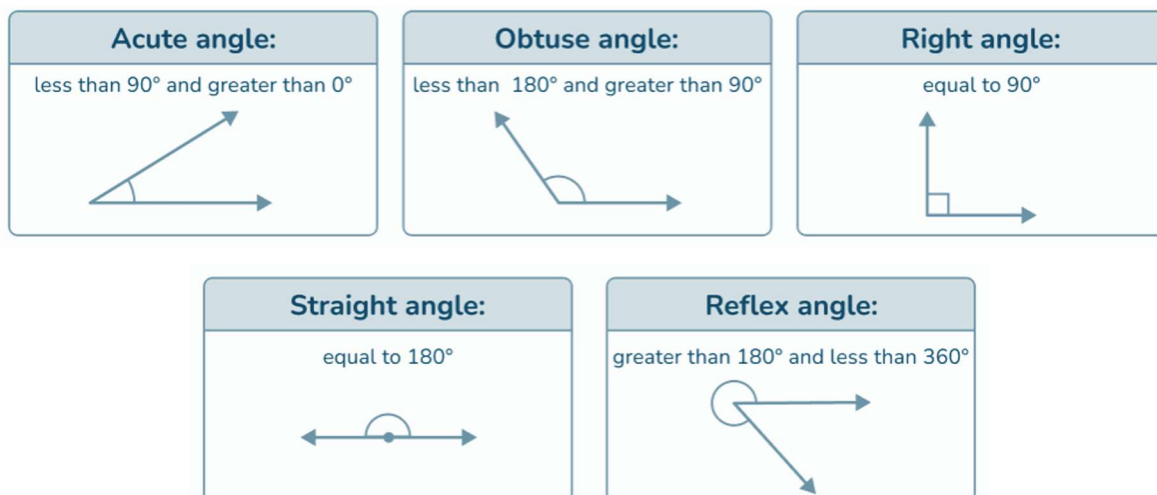


Figure 3: Types of Angles

In this task, the application should function as follows:

- Ask the user to enter an angle measurement for angle between 0 and 360 (NOT including 0 and 360), using only multiples of 10 degrees for simplicity.

- Display a visual representation of the angle based on the entered value.
- Prompt the user to identify the type of angle (e.g., acute, right, obtuse, reflex).
- If all 3 attempts are incorrect, the application should display the correct type.
- Process continues until the user has **practised** all 4 types of angles OR clicks on the “Home” button.

Task 3: Area Calculation of Shapes

[Basic level scoring]

Figure 4 shows the key shapes users must learn which are **Rectangle, parallelogram, triangle and trapezium**. In this task, the application should function as follows:

- Allow the user to select one of the four shapes.
- The application generates random numbers (1 to 20) for the appropriate input values (e.g. length and width for a rectangle) and displays them on the screen.
- The user has 3 minutes to calculate the area and enter their answer.
- Once the user enters the correct answer or uses up 3 unsuccessful attempts, the application should display the shape with labelled dimensions (e.g., length, width, height etc.) directly on the diagram, along with the formula and substituted values (see **Figure 4**).
- Process continues until the user has practised all 4 shapes OR clicks on the “Home” button.

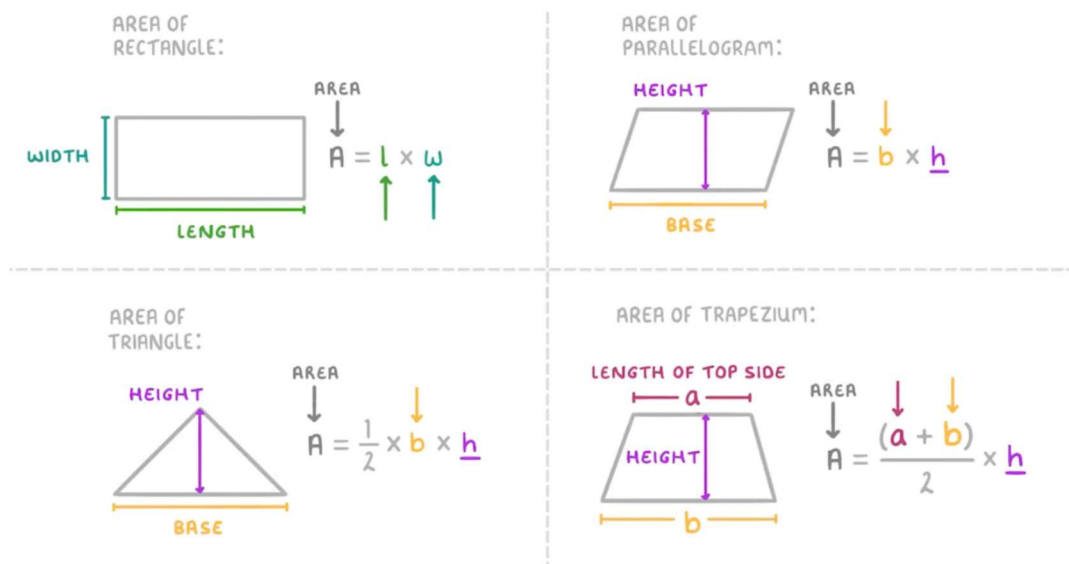


Figure 4: Area calculation of Different Shapes

Task 4: Area and Circumference Calculation of Circle

[Basic level scoring]

To help users learn better, the application should provide a guided and interactive experience of area/circumference calculations. An overview of the relevant formulas and methods is shown in **Figure 5** (Area and Circumference based on Radius) and **Figure 6** (Area and Circumference based on Diameter).

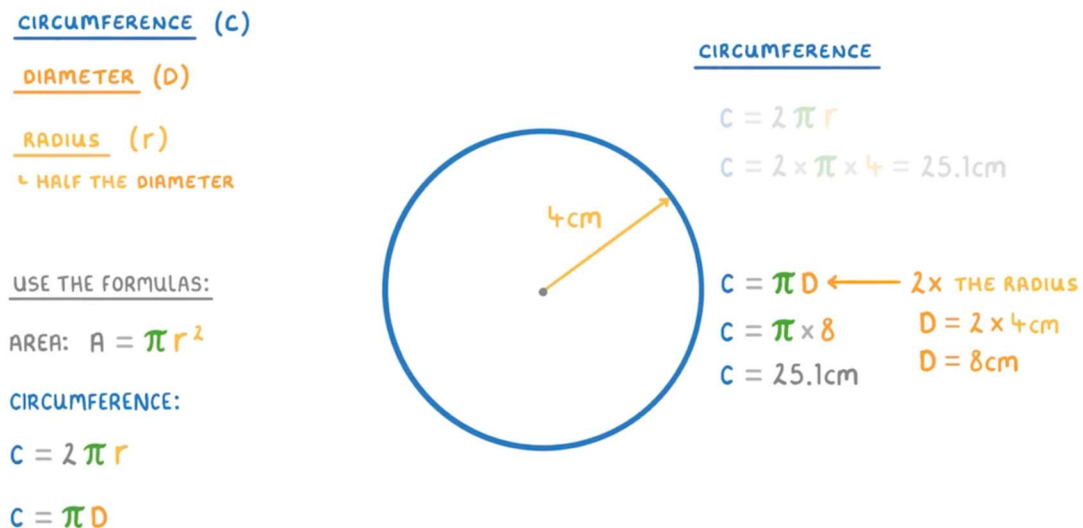


Figure 5: Calculating Area and Circumference based on Radius

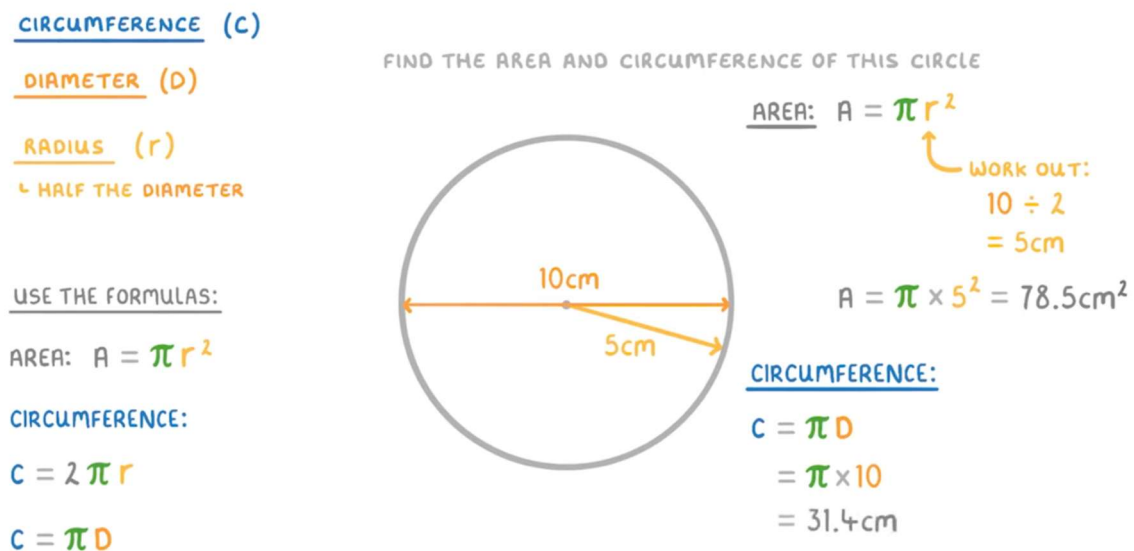


Figure 6: Calculating Area and Circumference based on Diameter

In this task, the application should function as follows:

- Allow the user to select either "Area" or "Circumference."
- Display either the radius or the diameter of a circle (random number between 1 and 20).
- The user must use the given value to calculate and enter the correct area or circumference.
- The user has 3 minutes to calculate and enter their answer.
- Once the user enters the correct answer or uses up 3 unsuccessful attempts, the application should display a circle with the radius/diameter labelled directly on the diagram, along with the formula and substituted values (see **Figure 5/6**).
- Process continues until the user has practised area and circumference calculations with **either** radius **or** diameter, OR clicks on the "Home" button.

Bonus 1: Compound Shapes Area Calculation

[Advanced Level scoring]

The methods for calculating the area of compound shapes are illustrated in **Figure 7**, **Figure 8**, and **Figure 9**.

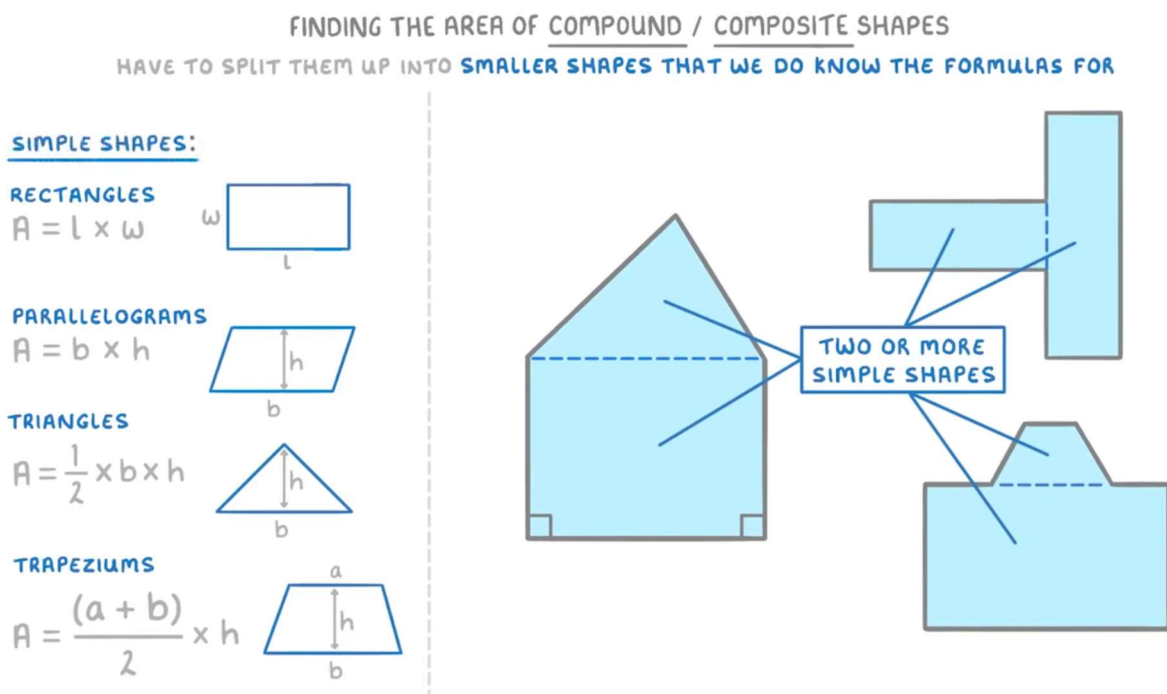


Figure 7: Calculating Area of Compound Shapes

SIMPLE SHAPES:

RECTANGLES

$$A = l \times w$$



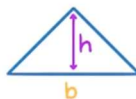
PARALLELOGRAMS

$$A = b \times h$$



TRIANGLES

$$A = \frac{1}{2} \times b \times h$$



TRAPEZIUMS

$$A = \frac{(a + b)}{2} \times h$$

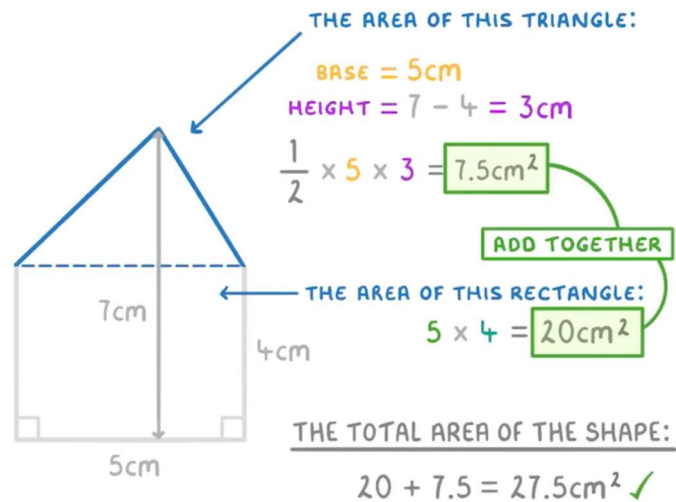
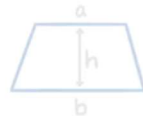


Figure 8: Calculating Area of Compound Shapes – Example 1

SIMPLE SHAPES:

RECTANGLES

$$A = l \times w$$



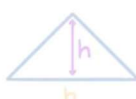
PARALLELOGRAMS

$$A = b \times h$$



TRIANGLES

$$A = \frac{1}{2} \times b \times h$$



TRAPEZIUMS

$$A = \frac{(a + b)}{2} \times h$$

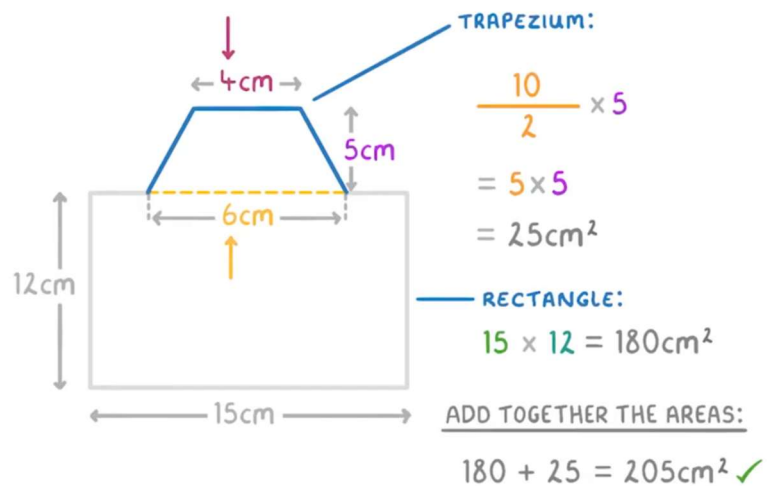
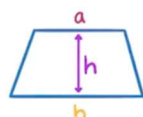


Figure 9: Calculating Area of Compound Shapes – Example 2

In this task, the program should function as follows:

- Allow the user to select one of the 6 compound shapes (Figure 10).
- Provide the respective values to calculate the area of the compound shapes.
- Allow 5 minutes to calculate the area and enter their answer.

Mini-Project: Shapeville v3

- After 3 incorrect attempts, application should display the correct formula and solution for the compound shape.
- Process continues until the user has practised all 6 compound shapes shown in **Figure 10** OR clicks the “Home” button.

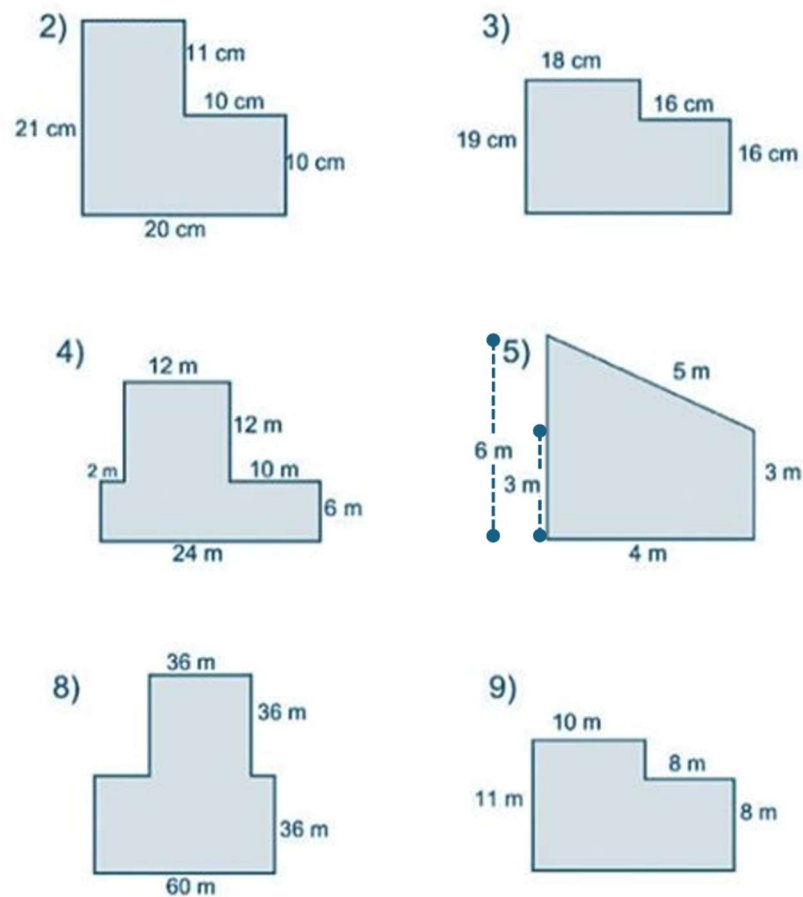


Figure 10: Compound Shapes for Area Calculation

Bonus 2: Sector of a Circle Area

[Advanced Level scoring]

The methods for calculating the area of a sector are demonstrated in **Figure 11** and **Figure 12**.

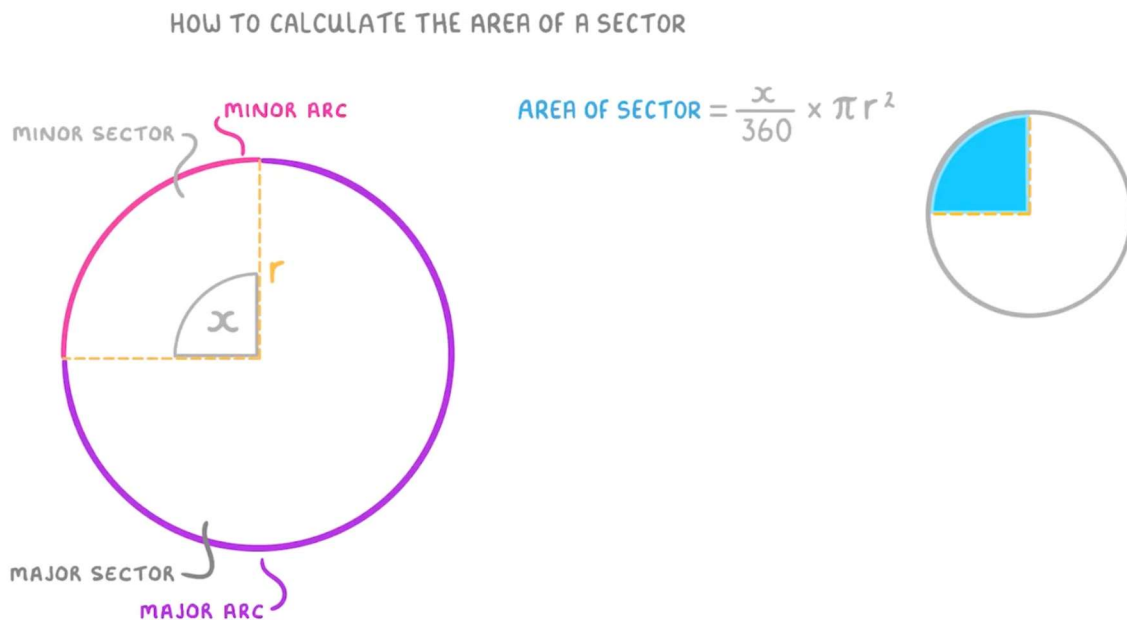


Figure 11: Calculating Area of a Sector of Circle

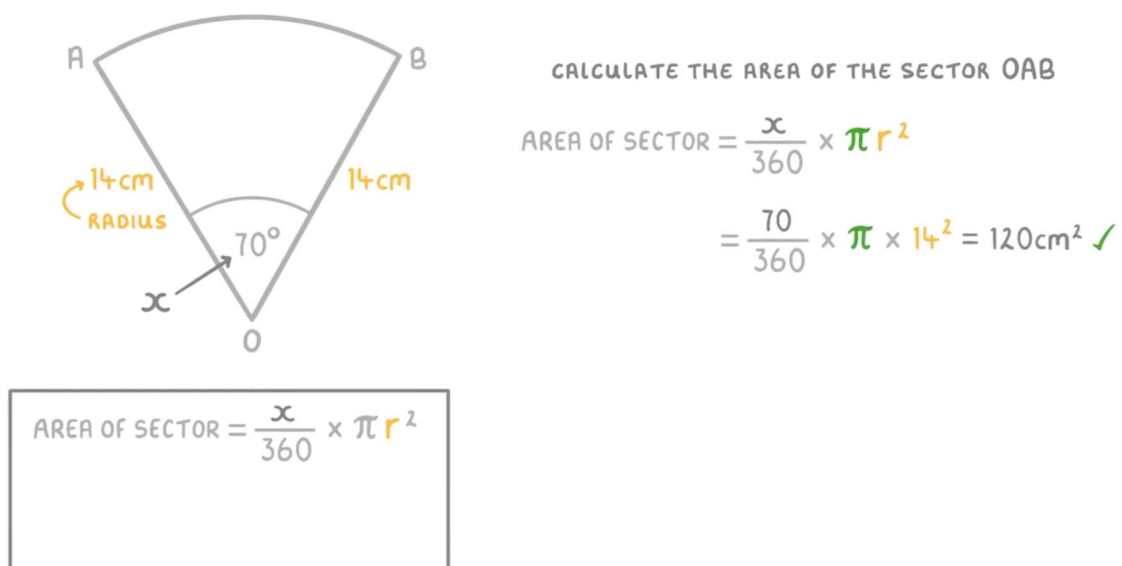
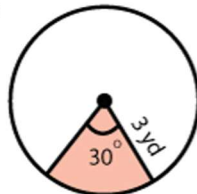


Figure 12: Calculating Area of a Sector of Circle – Example

In this task, the application should function as follows:

- Allow the user to select one of the 8 options of a sector of a circle as shown in **Figure 13**.
- Once selected, provide the user with the necessary values to calculate the area of the sector.
- Allow 5 minutes to calculate the area and enter their answer.
- After 3 incorrect attempts, application should display the correct formula and solution for the area of sector.
- Process continues until the user has practised all 8 options shown in **Figure 13** OR clicks the “Home” button.

Solved Example :



$$\begin{aligned}\text{Area of a sector} &= \frac{\text{central angle}}{360^\circ} \times \pi \times (\text{radius})^2 \\ &= \frac{Q \times \pi \times r^2}{360^\circ} \\ &= \frac{30^\circ \times 3.14 \times 3 \times 3}{360^\circ} = 2.35 \text{ yd}^2\end{aligned}$$

Find the area of the shaded region. Round the area to two decimal places (use $\pi = 3.14$)

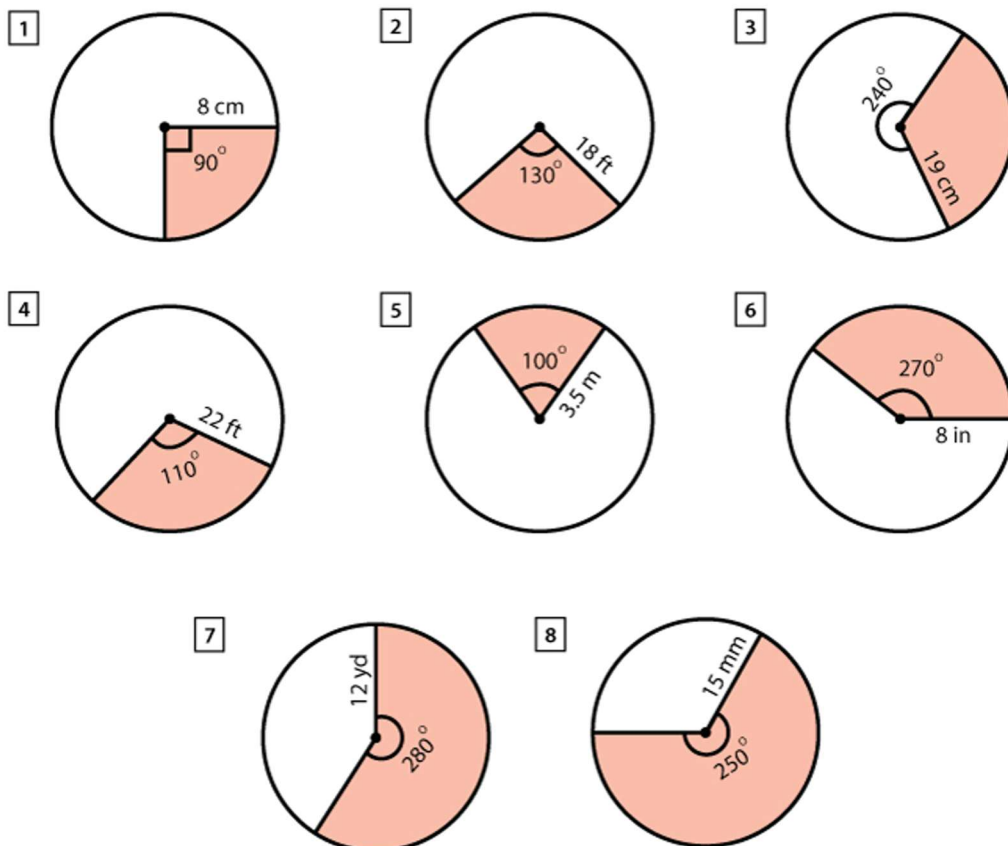


Figure 13: Examples for Area of a Sector Calculation

Mini-Project – Submission Requirements

Submit a zip file with your group number (e.g. “Group 1.zip”) containing items 1-3 below.

1) PDF report with the following contents:

- a) Allocation of tasks to each member of the group (1 page).
- b) Design of the system including flow charts and class diagrams (2-4 pages).
- c) Justification of your design choices, e.g. why were certain buttons or layouts chosen, how does your design help users complete the tasks easily paying attention to inclusivity and accessibility requirements (1/2 page – 1 page).
- d) User-manual for your application (1-2 pages).
- e) Description of the utilisation of AI and other tools which must include details of which specific tools have been used in which part of the Mini-Project (1-2 pages). THREE key prompts submitted to AI tools and related responses must be provided in the Appendix in a clearly organised manner, including a brief note on whether those responses were used as they were or modified by you. See Note-1 below for details.
- f) Reflection on your experience of the Mini-Project, as a group (1/2 a page). This must be an account of your genuine view, including what you think went well and not so well, how you might improve things if you were to do this project again etc. AI tools must NOT be used for this task.

2) All Java source files (with appropriate internal comments).

3) HTML javadocs.

Note-1: Mini-Project Policy for Using Generative AI Tools:

You are allowed, and encouraged, to use Generative AI for the Mini-Project. When using AI tools to generate code, the prompts should clearly reflect the tasks or problems you want to solve. Ensure your prompts illustrate the thought process behind using the AI tool and how it aided in the development of your project, building on your own knowledge and skills gained throughout the module. An example of an unacceptable use of an AI tool would be, “Write a Java program for a shape GUI application”.

Note-2: Tips for PDF report:

Organise your text clearly to maximise readability, e.g. use sub-headings, bullet-points etc. and minimise lengthy paragraphs. Font sizes should be reasonable (e.g. 11-12).

Note-3: Submission on QMPlus:

You are allowed to re-submit an unlimited number of times until the Cut-off-date; however, any submissions done after the Deadline will incur late penalties even if you had previously submitted before the deadline. Hence, it's advisable NOT to resubmit beyond the deadline unless absolutely necessary.

Note-4: Demonstration at mini-viva:

During the mini-viva, your submitted source code will be compiled and run on the assessor's computer (i.e. not your computer) in JDK 21.0.2 using the command line. The files you have provided must work as they are, without having to make any adjustments such as move or rename files/folders. If your application fails to run this way, you will lose all marks allocated for functionality.