**ARTIFICIAL INTELLIGENCE DEPARTMENT**

# Total Marks:

**Obtained Marks:**

**PROJECT NO#1**

FUNDAMENTAL OF PROGRAMMING

(LAB)

**Shape Calculator**

# Submitted To: Sir Shahid Raza

**Student Name:** **M.Muneeb, Zain ul Abdin Abbasi, M.Umar Abid, Mudassair**

**Section Name:** **BSAI-1A**

# Reg. Number: 22108190,22108249,22108188,22108251

**ARTIFICIAL INTELLIGENCE DEPARTMENT**

**Introduction: -**

The Shape Calculator program is a software application designed to assist users in performing various calculations related to geometric shapes. It is a handy tool for students, teachers, engineers, architects, and anyone who needs to work with shapes regularly.

The program is intuitive and easy to use, with a user-friendly interface that allows users to input the dimensions of a shape and calculate its area, perimeter and other relevant properties.

Some of the shapes that the Shape Calculator program can calculate include circles, rectangles, squares, Parallelograms and triangles. It also provides formulas and step-by-step instructions for each calculation, making it a valuable learning tool for students.

Overall, the Shape Calculator program is a powerful tool that simplifies complex calculations and helps users save time and effort. Whether you are a student, a professional, or just someone who needs to work with shapes regularly, this program is a must-have.

**Procedure: -**

The procedure for using the Shape Calculator program can vary slightly depending on the specific features and functions of the software, but here is a general overview of the steps involved:

1. Launch the Shape Calculator program on your computer or device.
2. Start by including the necessary header files for the program such as iostream, cmath, etc.
3. Select the shape you want to calculate from the list of options.
4. Input the required dimensions or measurements for the shape, such as the radius or diameter of a circle, the length and width of a rectangle, or the height and base of a triangle.
5. Press Enter button on keyboard.
6. The program will display the result of the calculation, such as the area, perimeter of the shape.
7. If desired, you can adjust the dimensions and recalculate the values to see how different measurements affect the outcome.
8. Some Shape Calculator programs may also offer additional features such as the ability to save or print the calculations, access formulas or tutorials, or perform more advanced calculations.

It's important to note that the specific procedure may vary depending on the particular Shape Calculator program being used, so it's always a good idea to refer to the program's user manual or documentation for more detailed instructions. Additionally, some programs may require specific inputs or measurements, so it's important to double-check the requirements before beginning any calculations.

**Objective: -**

The objective of a program that calculates the area and perimeter of different shapes such as circles, rectangles, squares, parallelograms, and triangles in C++ is to provide a tool that can perform these calculations quickly and accurately.

This program should be able to take user input for the necessary measurements, such as the radius of a circle or the lengths of the sides of a rectangle, and use the appropriate formulas to calculate the area and perimeter of the shape.

In addition to performing the calculations, the program should also display the results to the user in a clear and concise manner. This could be done by printing the values to the console or by displaying them in a graphical user interface.

**Scope of Shape Calculator: -**

The scope of a program that calculates the area and perimeter of different shapes such as circles, rectangles, squares, parallelograms, and triangles in C++ is fairly wide. It can be used in various fields such as mathematics, engineering, architecture, construction, and many others.

This program can be used by students, teachers, and professionals to quickly calculate the area and perimeter of different shapes without having to perform manual calculations. It can also be integrated into other programs or applications that require geometric calculations.

The program can be expanded to include additional shapes or to calculate other properties such as volume or surface area. It can also be adapted to work with different units of measurement such as inches, meters, or feet.

Overall, the scope of a program that calculates the area and perimeter of different shapes is quite broad and can be useful in a variety of contexts.

**Future Scope of Shape Calculator: -**

The future scope of a program that calculates the area and perimeter of different shapes such as circles, rectangles, squares, parallelograms, and triangles in C++ is quite broad. Here are some potential areas of future development:

1. *Integration with other software:* The program can be integrated with other software or applications, such as CAD software or 3D modeling software, to provide a complete solution for geometric calculations.
2. *Additional shapes:* The program can be expanded to include additional shapes such as polygons, cones, cylinders, and spheres.
3. *Graphical user interface:* The program can be developed with a graphical user interface (GUI) to make it more user-friendly and intuitive.
4. *3D calculations:* The program can be expanded to calculate the volume and surface area of 3D shapes such as prisms and pyramids.
5. *Unit conversion:* The program can be developed to allow the user to input measurements in various units and then convert them to a standardized unit for calculation.

Overall, there are many potential areas for future development of a program that calculates the area and perimeter of different shapes in C++. These developments can make the program more versatile, user-friendly, and applicable to a wider range of use cases.

**Features of Shape Calculator: -**

Here are some features that a program that calculates the area and perimeter of different shapes such as circles, rectangles, squares, parallelograms, and triangles in C++ may have:

1. *Shape selection:* The program allows the user to select the shape they want to calculate the area and perimeter for, such as circle, rectangle, square, parallelogram, or triangle.
2. *Calculation:* The program performs the necessary calculations to determine the area and perimeter of the selected shape based on the user's input.
3. *Output:* The program displays the calculated area and perimeter values for the selected shape in a clear and easy-to-understand format.
4. *User-friendly interface:* The program has a user-friendly interface that makes it easy for the user to input the necessary values and obtain the calculated results.
5. *Modularity:* The program is designed in a modular way so that additional shapes can be easily added in the future.

Overall, these features make a program that calculates the area and perimeter of different shapes in C++ more useful and user-friendly, and allow it to be easily extended and adapted to different use cases.

**Disadvantages of Shape Calculator: -**

Here are some potential disadvantages of a program that calculates the area and perimeter of different shapes such as circles, rectangles, squares, parallelograms, and triangles:

1. *Limited functionality:* The program is limited to calculating the area and perimeter of specific shapes, and may not be useful for more complex geometric calculations.
2. *Input errors:* The program relies heavily on user input, and if the user inputs incorrect values, the calculations will be incorrect.
3. *Lack of real-world application:* While the program is useful for mathematical and scientific calculations, it may not have a direct real-world application for the average user.
4. *Not visually appealing:* The program may lack a visually appealing interface, making it less engaging for users.
5. *Limited platform compatibility:* The program may only be compatible with certain operating systems or hardware, limiting its accessibility.
6. *Lack of customization:* The program may not allow for customization, such as changing the colors or font size of the interface.

Overall, while a program that calculates the area and perimeter of different shapes in C++ can be useful, it may have limitations and drawbacks that make it less effective for certain use cases or users.

**Requirements: -**

Here are some requirements that a program that calculates the area and perimeter of different shapes such as circles, rectangles, squares, parallelograms, and triangles should meet:

1. The program should allow the user to select the shape they want to calculate the area and perimeter for.
2. The program should validate user input to ensure that it is a valid value for the selected shape.
3. The program should perform the necessary calculations to determine the area and perimeter of the selected shape.
4. The program should display the calculated area and perimeter values for the selected shape in a clear and easy-to-understand format.
5. The program should have a user-friendly interface that makes it easy for the user to input the necessary values and obtain the calculated results.
6. The program should be modular, so that additional shapes can be easily added in the future.
7. The program should be able to handle errors and exceptions that may occur during the calculation process, such as division by zero or overflow.
8. The program should be efficient and not use unnecessary resources, such as memory or processing power.
9. The program should be platform-independent and compatible with different operating systems and hardware.
10. The program should have adequate documentation and user support to assist users in its operation and troubleshooting.

Overall, these requirements ensure that the program is functional, user-friendly, efficient, and adaptable to different use cases and environments.

**Code: -**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int choice;

cout<<"SHAPE CALCULATOR"<<endl;

cout << "Enter the shape (1 = circle, 2 = rectangle, 3 = square, 4 = parallelogram, 5 = triangle): ";

cin >> choice;

switch (choice)

{

case 1:

double radius, area\_c, perimeter\_c;

cout << "Enter the radius of the circle: ";

cin >> radius;

area\_c = M\_PI \* pow(radius, 2);

perimeter\_c = 2 \* M\_PI \* radius;

cout << "Area of the circle: " << area\_c << endl;

cout << "Perimeter of the circle: " << perimeter\_c << endl;

break;

case 2:

double length, width, area\_r, perimeter\_r;

cout << "Enter the length of the rectangle: ";

cin >> length;

cout << "Enter the width of the rectangle: ";

cin >> width;

area\_r = length \* width;

perimeter\_r = 2 \* (length + width);

cout << "Area of the rectangle: " << area\_r << endl;

cout << "Perimeter of the rectangle: " << perimeter\_r << endl;

break;

case 3:

double side, area\_s, perimeter\_s;

cout << "Enter the length of the side of the square: ";

cin >> side;

area\_s = pow(side, 2);

perimeter\_s = 4 \* side;

cout << "Area of the square: " << area\_s << endl;

cout << "Perimeter of the square: " << perimeter\_s << endl;

break;

case 4:

double base, height, area\_p, perimeter\_p, length\_p;

cout << "Enter the base of the parallelogram: ";

cin >> base;

cout << "Enter the height of the parallelogram: ";

cin >> height;

cout << "Enter the length of the parallelogram: ";

cin >> length\_p;

area\_p = base \* height;

perimeter\_p = 2 \* (base + length\_p);

cout << "Area of the parallelogram: " << area\_p << endl;

cout << "Perimeter of the parallelogram: " << perimeter\_p << endl;

break;

case 5:

double base\_t, height\_t, area\_t, perimeter\_t, side1\_t, side2\_t, side3\_t;

cout << "Enter the base of the triangle: ";

cin >> base\_t;

cout << "Enter the height of the triangle: ";

cin >> height\_t;

cout << "Enter the length of the side one of the triangle: ";

cin >> side1\_t;

cout << "Enter the length of the side Two of the triangle: ";

cin >> side2\_t;

cout << "Enter the length of the side Three of the triangle: ";

cin >> side3\_t;

area\_t = 0.5 \* base\_t \* height\_t;

perimeter\_t = side1\_t + side2\_t + side3\_t;

cout << "Area of the triangle: " << area\_t << endl;

cout << "Perimeter of the triangle: " << perimeter\_t << endl;

break;

default:

cout << "Invalid choice!" << endl;

break;

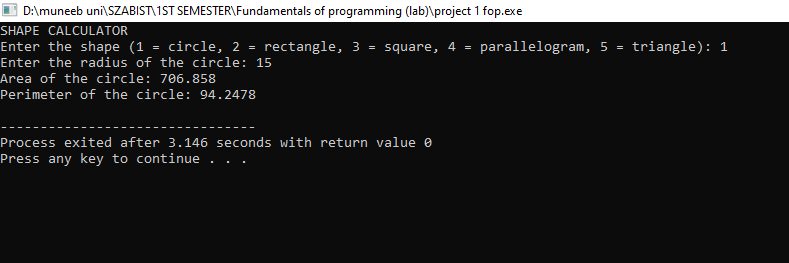
}

return 0;

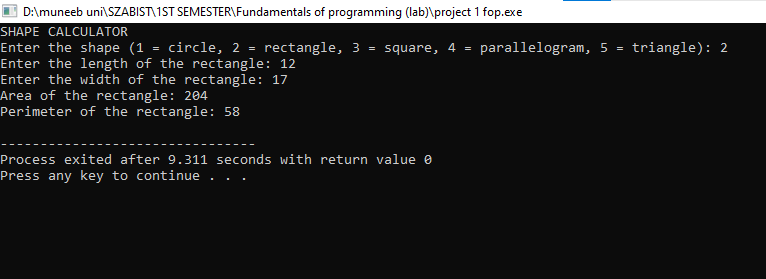
}

**Output: -**

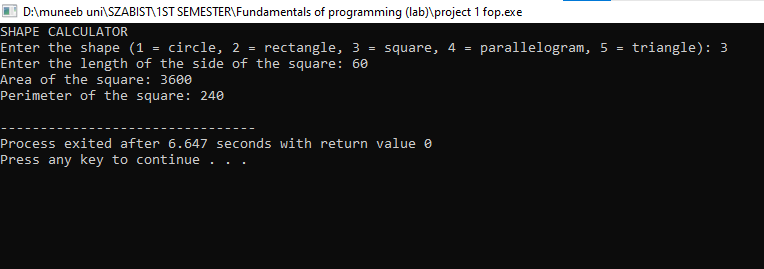
**1: -circle**

****

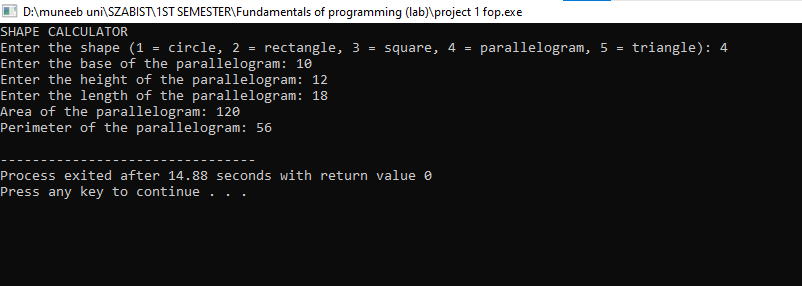
**2: -Rectangle**

****

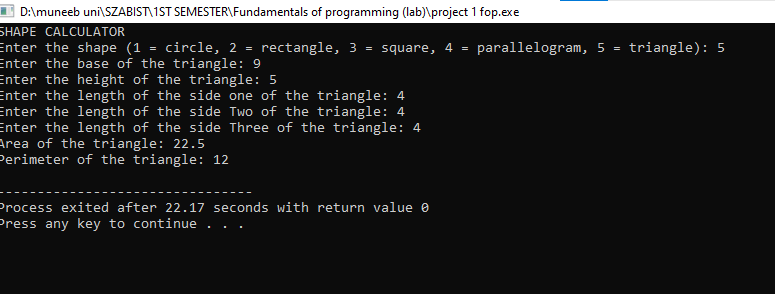
**3: -Square**

****

**4: -parallelogram**

****

**5: -Triangle**

****

***THE END***