1. Escribir un programa en C++ con una clase ***Circulo*** que tenga como atributos el ***radio*** y el ***área*** del círculo. Además, debe tener los siguientes métodos:
   - Constructor por defecto con radio que valga 10
   - Constructor con inicializadores
   - Constructor con parámetros
   - Constructor copia
   - Los métodos ***get*** y ***set*** para obtener y dar valor a los atributos
   - Método para calcular el valor de área en función del radio.

   El programa creará cuatro objetos, ***C1, C2, C3*** y ***C4*** utilizando los constructores anteriores. Se debe conseguir tener círculos de 10, 5 y 15 cm de radio, calcular el área para cada uno de ellos y mostrar los datos de cada uno por pantalla.

2. Escribir un programa en C++ con una clase ***vectorElementos*** que tenga un atributo que sea un ***array*** de 10 enteros.

   La clase debe permitir modificar los valores de los elementos del ***array*** mediante un método ***cambiaElemento*** y calcular la suma de los elementos del ***array*** mediante un método ***calculaSuma***.

   Además, cuando se crea un objeto de esta clase el ***array*** contendrá los valores del 1 al 10.

   Después, implementar un programa en C++ que cree un objeto de esta clase y muestre por pantalla el resultado de la suma de los elementos del ***array***, antes de modificarlos y después de hacer varias modificaciones.

   Repetir el proceso de modificación de valores mientras quiera el usuario.

3. Realizar un programa en C++ que contenga una clase llamada ***CalcularNumPositivo***.

   La clase tiene dos atributos, ***número*** y ***resultado***, ambos de tipo decimal y los métodos:

   - Constructores por defecto y por copia.
   - ***Set*** y ***Get*** necesarios para manejar los atributos.

- ***sumaNum***, método que recibe el valor de un número que suma a número. Guarda la suma realizada en el atributo resultado y devuelve dicho valor para mostrarlo en pantalla desde el ***main***.
- ***restaNum***, método que recibe el valor de un número que resta a número. Guarda la resta realizada en el atributo resultado y devuelve dicho valor para mostrarlo en pantalla desde el ***main***.
- ***multiplicaPor***, método que recibe el valor de un número que multiplica por número. Guarda el producto realizado en el atributo resultado y devuelve dicho valor para mostrarlo en pantalla desde el ***main***.
- ***dividePor***, método que recibe el valor del denominador para dividir por número. Guarda la división realizada en el atributo resultado y devuelve dicho valor para mostrarlo en pantalla desde el ***main***. Además, debe lanzar dos excepciones (tipo ***string***), una si la división es por 0 y otra si se divide por un número negativo.

4. Realizar la sobrecarga de otras operaciones binarias y unarias sobre la clase ***Rectángulo***: resta, decremento, **>**, **<**, **==**.

5. Definir la clase ***Complejo*** con dos atributos reales, parte real y parte imaginaria, que permita declarar números complejos (objetos de la clase Complejo) como z1 = (a, b) y z2 = (c, d), donde a y b son la parte real y b y d la parte imaginaria de los números z1 y z2, respectivamente.

   Declarar e implementar los constructores y métodos/funciones ***get/set*** necesarios para acceder a sus variables.

   Finalmente, definir e implementar operadores sobrecargados para realizar las siguientes operaciones binarias y unarias miembros de la clase ***Complejo***:

   - Suma: $z1+z2=(a+c, b+d)$
   - Diferencia: $z1 - z2=(a - c, b - d)$
   - Producto: $z1\ z2=(ac - bd, ad+bc)$
   - División: $Z1/Z2 = ((ac + bd)/(c^2 + b^2),(bc - ad)/(c^2 + b^2))$
   - Pre Incremento: $++z1=(a+1, b+1)$
   - Pre Decremento: $--z1=(a - 1, b - 1)$
   - Post Incremento como en C++: $z1{+}{+} =(a+1, b+1)$
   - Post Decremento como en C++: $z1{-}{-} =(a-1, b-1)$

- Operador de extracción **>>**
- Operador de inserción **<<**

Repetir los operadores anteriores (incremento, decremento, **<<** y **>>**) como funciones generales y declaradas como operadores ***friend*** dentro de la clase ***Complejo***.

6. Realizar una calculadora que, a través de funciones templatizadas, permita realizar la suma, resta, multiplicación y división de diferentes tipos de datos: enteros, decimales, complejos.

7. Crear una clase ***Cpares*** que tenga dos atributos, una clave y un valor. Los tipos de ambos atributos deben estar templatizados.

   Crear una clase templatizada ***Vpares*** que contenga un vector de ***Cpares***.

   Sobrecargar el operador >> para ***Cpares*** y ***Vpares***, de modo que se puedan mostrar por pantalla usando ***cout << var***

   Crear un programa ***main*** que realice lo siguiente:

   - Solicitar 5 datos para crear un objeto de ***Vpares*** donde sus elementos sean de tipo: ***string, int***.
   - Solicitar 5 datos para crear un objeto de ***Vpares*** donde sus elementos sean de tipo: ***float, int***.
   - Mostrar los correspondientes resultados por pantalla usando el operador **<<**.

8. Realizar un programa con tres clase:

   Clase abuelo:

   - 3 atributos de tipo i***nt***: ***private***, ***protected*** y ***public***
   - Método: ***void iniciarTodo();***

   Clase padre con diferentes accesos de herencia a la clase ***abuelo***: ***private***, ***protected*** y ***public*** y el método i***niciarTodo*** para dar valor a los atributos.

   Clase nieto con acceso de herencia ***public*** a la padre y el método i***niciarTodo*** para dar valor a los atributos.

   El ***main*** debe declarar un objeto para cada clase. Invocar al método ***iniciarTodo*** y mostrar por pantalla el valor que toman los atributos.

9. Realizar un programa que responda al diagrama de la figura y que presente un menú con las siguientes opciones: (1) Dar de alta cuadrado; (2) Dar de alta triangulo; (3) Mostrar poligonos regulares y (0) SALIR.

   Los polígonos se deberán almacenar en un vector. Los métodos son:

   - Método ***darAlta***, debe permitir al usuario introducir los atributos de cada ***clase.***
   - Método ***getNombre***, según corresponda, devolverá un ***string*** con el nombre de cada polígono regular (cuadrado, triángulo, etc.)
   - Los métodos ***gets*** restantes permitirán obtener el valor de los atributos correspondientes a cada clase.