

# Module

Exceptions

# Exceptions

```
decimal number = decimal.Parse(Console.ReadLine());  
  
decimal division = 1 / number;  
Console.WriteLine($"Ett delat med talet blir {division}");
```

# Exceptions

```
try
{
    decimal number = decimal.Parse(Console.ReadLine());

    decimal division = 1 / number;
    Console.WriteLine($"Ett delat med talet blir {division}");
}
```

# Exceptions

```
try
{
    decimal number = decimal.Parse(Console.ReadLine());

    decimal division = 1 / number;
    Console.WriteLine($"Ett delat med talet blir {division}");
}
catch (DivideByZeroException)
{
    Console.WriteLine("Division med 0");
}
catch (FormatException)
{
    Console.WriteLine("Fel i formatet");
}
```

# Exceptions

```
private static double TriangleArea(double height, double width)
{

    return height * width / 2;
}
```

# Exceptions

```
private static double TriangleArea(double height, double width)
{
    if (width < 0 && height < 0)
        throw new ArgumentException("Både höjden och bredden är negativ!");

    if (height < 0)
        throw new ArgumentException("Höjden får inte vara negativ!");

    if (width < 0)
        throw new ArgumentException("Bredden får inte vara negativ!");

    return height * width / 2;
}
```