Errors in Programming

In programming, errors can generally be categorized into three main types: **syntax errors**, **runtime errors**, and **logical errors**. Understanding these error types is crucial for effective debugging and writing robust code. Let's explore each type with examples and how to handle them in C#.

1. Syntax Errors

Syntax errors occur when the code violates the grammatical rules of the programming language. These errors are usually detected by the compiler before the program runs.

Example:

```
int number = 10
Console.WriteLine(number);
```

```
string number = "10";
```

---

2. Runtime Errors

Runtime errors occur while the program is running, usually because of operations that are not logically valid at runtime, such as dividing by zero, accessing an array out of bounds, or attempting to use a null reference.

Example:

```
int divisor = 0;
int result = 10 / divisor;
Console.WriteLine(result);
```

How to Handle:

```
try
{
   int divisor = 0;
   int result = 10 / divisor;
   Console.WriteLine(result);
}
catch (DivideByZeroException ex)
{    Console.WriteLine("Error: " + ex.Message);}
```

OR

Validations:

```
int divisor = 0;
if (divisor != 0)
{
   int result = 10 / divisor;
   Console.WriteLine(result);
}
else
{
   Console.WriteLine("Cannot divide by zero.");
}
```

```
string message = null;
Console.WriteLine(message.Length);
```

➔ System.NullReferenceException: Object reference not set to an instance of an object.

```
string message = null;
if (message != null)
{
   Console.WriteLine(message.Length);
}
else
{
   Console.WriteLine("Message is null.");
}
```

3. Logical Errors

Logical errors occur when the program compiles and runs, but the output is not what you expected. These errors are the most difficult to detect because the program doesn't crash or show any immediate signs of error.

Example:

```
int[] numbers = {1, 2, 3, 4, 5};
for (int i = 0; i <= numbers.Length; i++) // Logical error: should be i < numbers.Length;
{
    Console.WriteLine(numbers[i]);
}
```

➔ System.IndexOutOfRangeException: Index was outside the bounds of the array.

```
for (int i = 0; i < numbers.Length; i++)
{
    Console.WriteLine(numbers[i]);
}
```

```
string input = "yes";
if (input == "Yes") // Logical error: comparison is case-sensitive
{
    Console.WriteLine("Input is yes");
}
else
{
    Console.WriteLine("Input is not yes");
}
```

➔ incorrect string comparison

```
if (input.Equals ("Yes", StringComparison.OrdinalIgnoreCase))
{  Console.WriteLine("Input is yes"); }
else
{   Console.WriteLine("Input is not yes"); }
```