

Reading/Fixing Compilation Errors



When compiling a program, often many errors can occur. An attempted compile cycle with many errors can be overwhelming, but approaching the problem in a systematic way will help wade through the error list. Below is an example which shows some errors that could occur when compiling a program.

```
:~$ javac CompileErrors.java
CompileErrors.java:1 error: '{' expected
public class CompileErrors
      ^
CompileErrors.java:5 error: ';' expected
    int x = 0
          ^
CompileErrors.java:6: error: ')' expected
    if(x == 0
        ^
CompileErrors.java:8: error: ')' expected
    System.out.println("Hello World");
                        ^
4 errors
```

Let's begin with looking at the first error. The first error statement is

```
CompileErrors.java:1: error: '{' expected
public class CompileErrors
```

Below shows the major components of an error statement:

1. The class/file where the error occurs
2. The line number where or around the error occurs
3. The type of error that occurs
4. The actual line of code and location of the error marked with a carat.

To summarize this first error statement, a curly brace is missing after the code found on line 1 in the CompileErrors.java file. Adding a curly brace after this line of code would fix this error.

Example 1:

```
:~$ javac CompileErrors.java
CompileErrors.java:17: error: reached end of file while parsing
    }
1 error
```

This error statement above can be read as an error has occurred at line 17 in the CompileErrors.java class/file where it reached the end of file while parsing. Generally, this error occurs when there's a curly brace issue, for example, if a curly brace was missing at the end of the file.

Example 2:

Reading/Fixing Compilation Errors



```
:~$ javac CompileErrors.java
CompileErrors.java:6: error: variable x might not have been initialized
    if(x == 0)
       ^
1 error
```

The error statement above can be read as an error is occurring at line 6 of the code, in the class `CompileErrors.java`, where the variable `x` might not have been initialized. This error occurs when the variable `x` has not been set to a value yet. It can not evaluate the expression, `if(x == 0)`, `x` contains no value.

Example 3:

```
:~$ javac CompileErrors.java
CompileErrors.java:14: error: cannot find symbol
    System.out.print(x + y);
                      ^
    symbol: variable y
    location: class CompileErrors
1 error
```

The error statement above can be read as at line 14 of the `CompileErrors.java` file, it cannot find the variable `y` in the `CompileErrors.java` file. This error occurs when the variable has not been declared.

Example 4:

```
:~$ javac CompileErrors.java
CompileErrors.java:11: error: incompatible types
    int hello = "Hello";
                ^
    required: int
    found:    String
1 error
```

The above error statement can be read as at line 11, of the `CompileErrors.java` file, there are incompatible types with the variable `hello`. This error occurs when attempting to set a certain type of variable to an incompatible type, for example, attempting to set a `String` to an `int` variable.