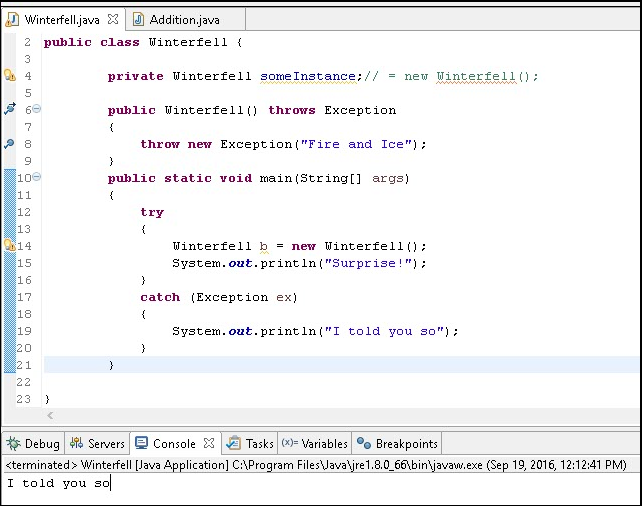
**Sameera Bammidi (G01025127)**

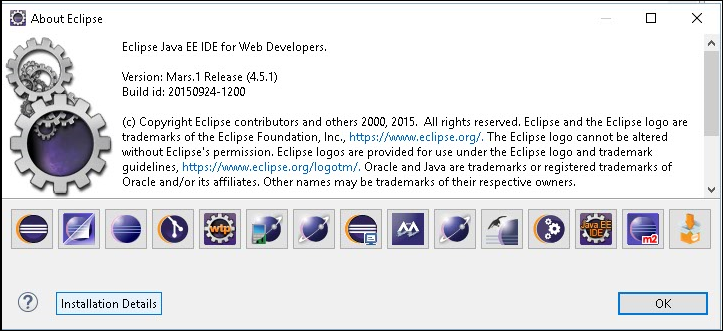
**SWE 619 Assignment 2**

1. Fixing the error:

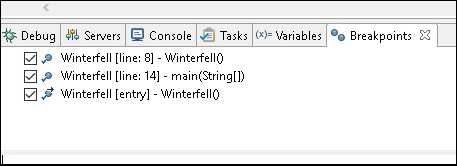
I commented the creation of a new private instance of the Winterfell object. And Executed the program. The program printed: I told you so (ScreenShot below )



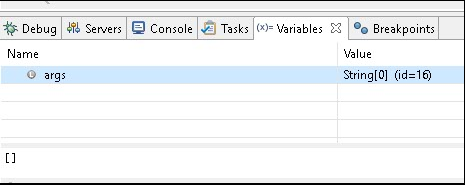
1. IDE Debugging and Version: Eclipse Java EE IDE for Web Developers. Version: Mars.1 Release (4.5.1) (Screenshots Below)



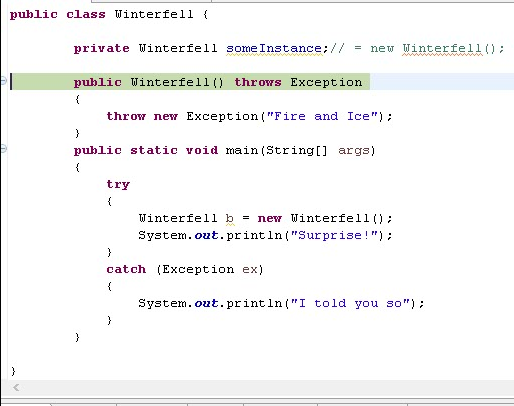
**Break Points:**

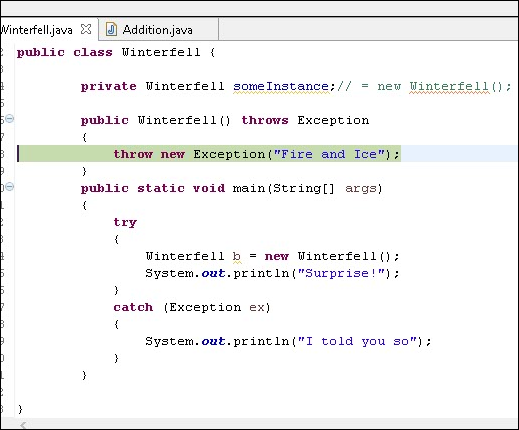


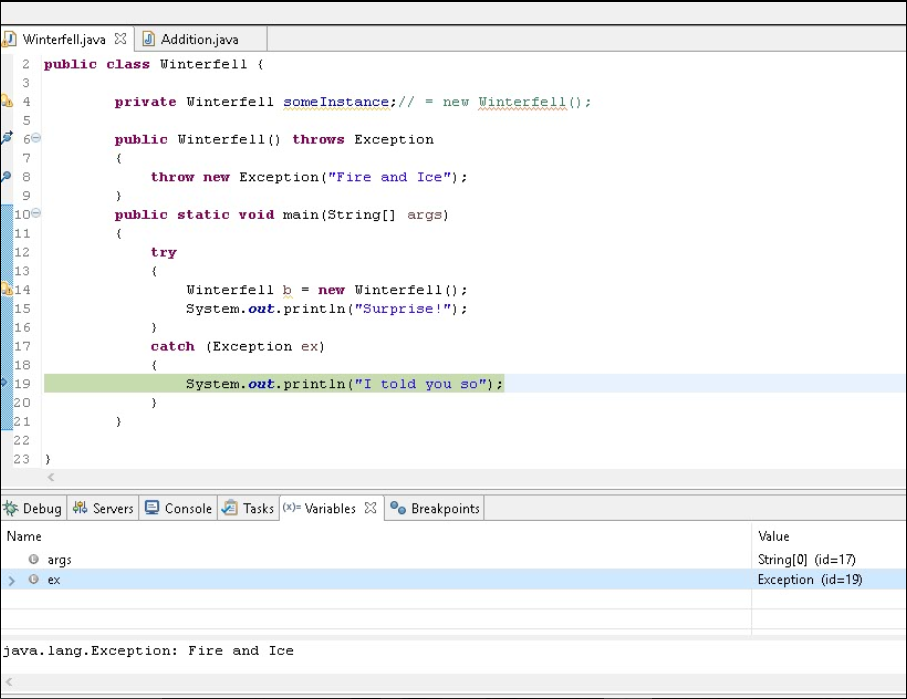
**Variables / State while Debugging:**



**ScreenShots highlighting the code as the debugger steps through it:**







1. When the *main* method is invoked:

Firstly, in the *try* block, a new instance of Winterfell class is explicitly created when it is instantiated and assigned to the variable ***b***. Space is allocated for the new ***b***.

Next, the Winterfell constructor with no arguments is first invoked.

Now the Java compiler invokes its superclass's (*Object*) constructor, which has no arguments (*Object() { }*). Now any instance variable or instance initializers are invoked and then the body of the no argument constructor is executed.

Next,

***In the given version***, there is a constructor call to initialize the instance variable which is also of the type Class Winterfell. This executes before the body of the constructor of the Class Winterfell executes. The instance variable initialization itself makes another call to create the object. This happens recursively and results in a stack overflow error. This happens because there is no more memory available on the stack. To avoid this infinite loop, we should not initialize by calling a constructor that throws Exception.

***In my modified version,*** there are no instance variable initializers in the Winterfell constructor (It will get initialized to null, though memory is allocated on the stack for it). So the body of the Winterfell constructor which begins with the “throw” statement is executed and returns to the main method. Since it throws an exception(Exception), the exception is caught in the *catch* block and executes its body(Prints: *I told you so*).  We do not see the “Surprise!” statement being printed because control is transferred to the catch block due to the exception.

Ref: **12.5. Creation of New Class Instances**

https://docs.oracle.com/javase/specs/jls/se7/html/jls-12.html