**Assignment 2**

**Rasika Mohod**

rmohod@gmu.edu

G01044774

**Original code:**

**public** **class** Winterfell {

**private** Winterfell someInstance = **new** Winterfell();

**public** Winterfell() **throws** Exception {

**throw** **new** Exception(“Fire and Ice”);

}

**public** **static** **void** main(String[] args){

**try**{

Winterfell b = **new** Winterfell();

System.out.println(“Surprise!”);

} **catch** (Exception ex){

System.out.println(“I told you so”);

}

}

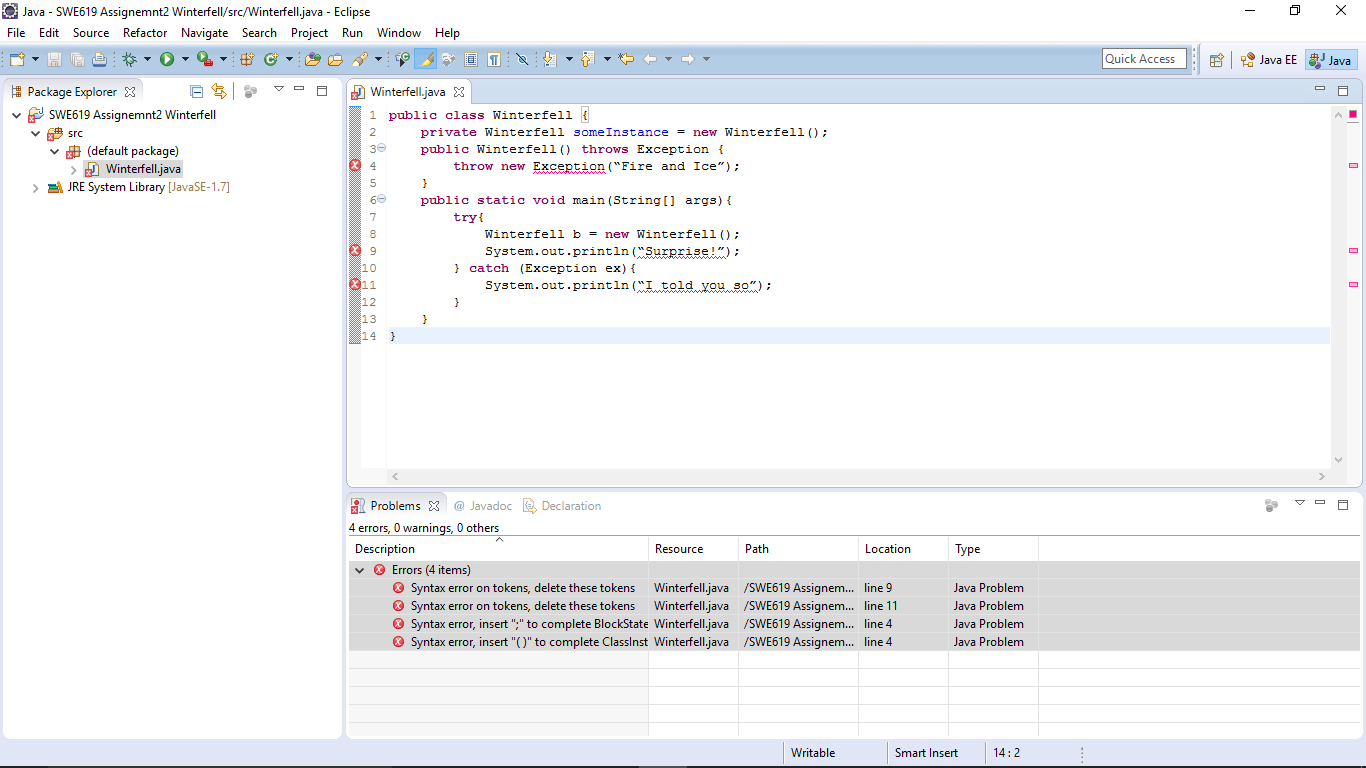
}

**a) Compile the program (fix any errors). Execute the program. What does the program print?**

**IDE used:** Eclipse (Luna)

**Errors before compilation:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Resource | Path | Location | Type |
| Syntax error on tokens, delete these tokens | Winterfell.java | /SWE619 Assignemnt2 Winterfell/src | line 9 | Java Problem |
| Syntax error on tokens, delete these tokens | Winterfell.java | /SWE619 Assignemnt2 Winterfell/src | line 11 | Java Problem |
| Syntax error, insert ";" to complete BlockStatements | Winterfell.java | /SWE619 Assignemnt2 Winterfell/src | line 4 | Java Problem |
| Syntax error, insert "( )" to complete ClassInstanceCreationExpression | Winterfell.java | /SWE619 Assignemnt2 Winterfell/src | line 4 | Java Problem |



**Errors after compilation (without any code change):**

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Syntax error on tokens, delete these tokens

Syntax error on tokens, delete these tokens

at Winterfell.main(Winterfell.java:9)

**Code changes done:**

Line 4:

Old: **throw** **new** Exception(“Fire and Ice”);

New: **throw** **new** Exception("Fire and Ice");

Changes: “” to ""

Line 9:

Old: System.out.println(“Surprise!”);

New: System.***out***.println("Surprise!");

Changes: “” to ""

Line 11:

Old: System.out.println(“I told you so”);

New: System.***out***.println("I told you so");

Changes: “” to ""

**Correct code:**

**public** **class** Winterfell {

**private** Winterfell someInstance = **new** Winterfell();

**public** Winterfell() **throws** Exception {

**throw** **new** Exception("Fire and Ice");

}

**public** **static** **void** main(String[] args){

**try**{

Winterfell b = **new** Winterfell();

System.***out***.println("Surprise!");

} **catch** (Exception ex){

System.***out***.println("I told you so");

}

}

}

**Run time error:**

Exception in thread "main" java.lang.StackOverflowError

at Winterfell.<init>(Winterfell.java:3)

at Winterfell.<init>(Winterfell.java:2)

**Line 2:**

**private** Winterfell someInstance = **new** Winterfell(); is not needed as it is giving Stack overflow error.

**Correct code:**

**public** **class** Winterfell {

//private Winterfell someInstance = new Winterfell();

**public** Winterfell() **throws** Exception {

**throw** **new** Exception("Fire and Ice");

}

**public** **static** **void** main(String[] args){

**try**{

Winterfell b = **new** Winterfell();

System.***out***.println("Surprise!");

} **catch** (Exception ex){

System.***out***.println("I told you so");

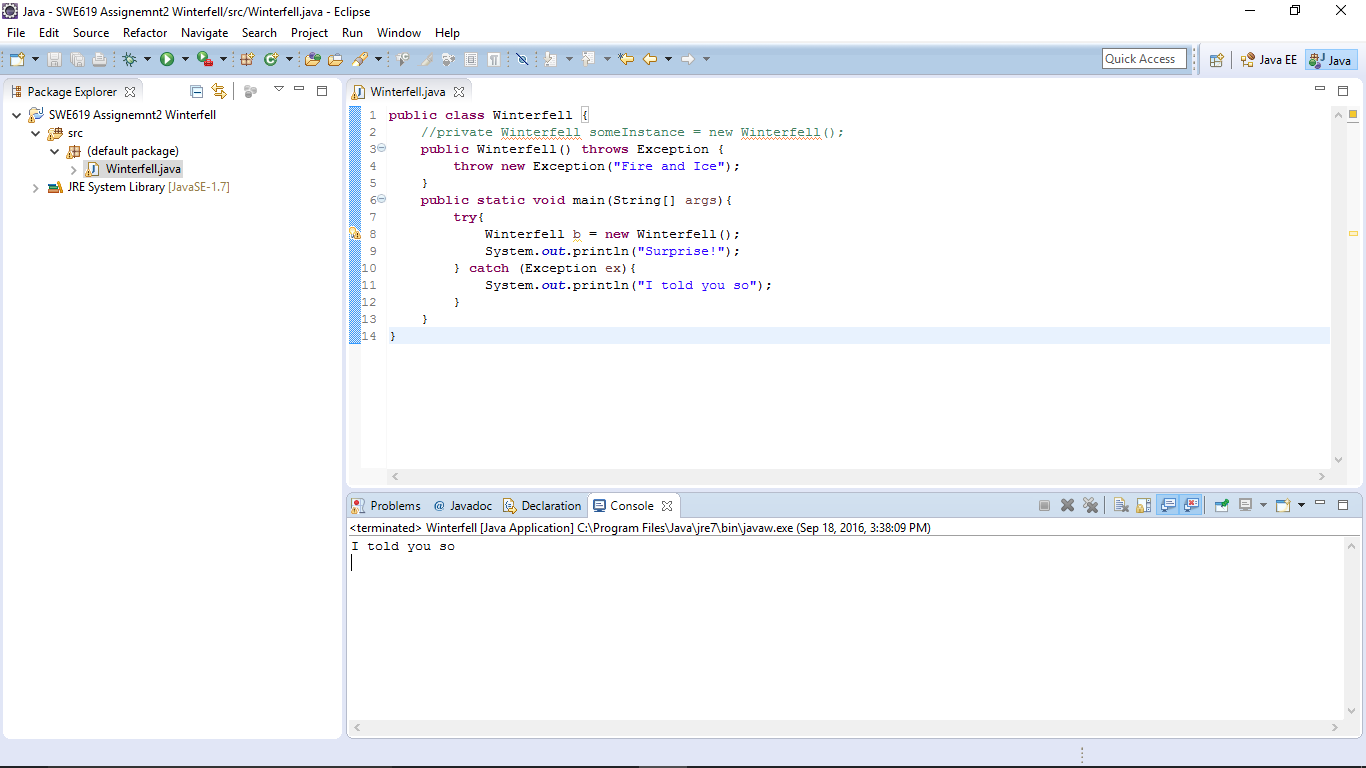
}

}

}

**Output after code run:**

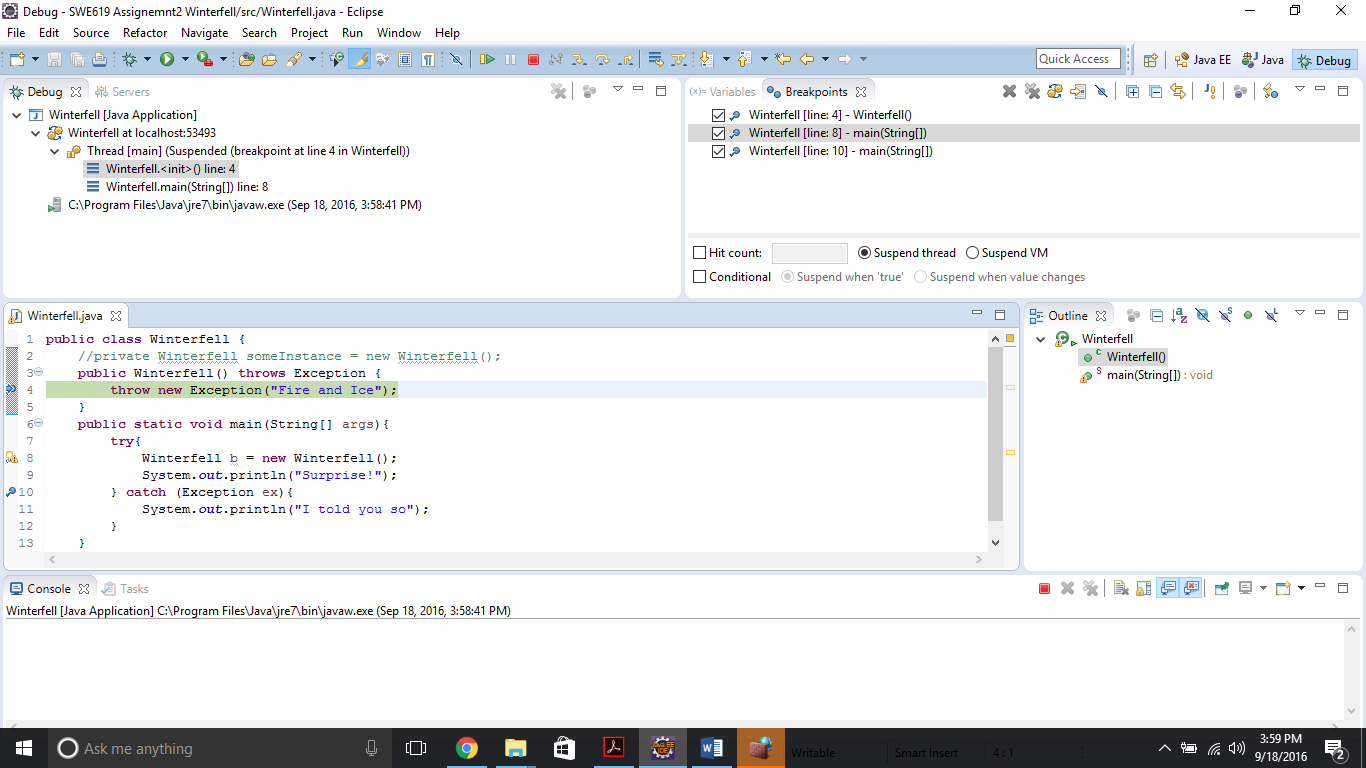
I told you so



**b) Use your IDE debugger to step through the program. Show some sample screen shots. Provide IDE name and version, some breakpoints, and variables with their associated values as you step through the program.**

**IDE: Eclipse (Luna)**

**Debug mode:**



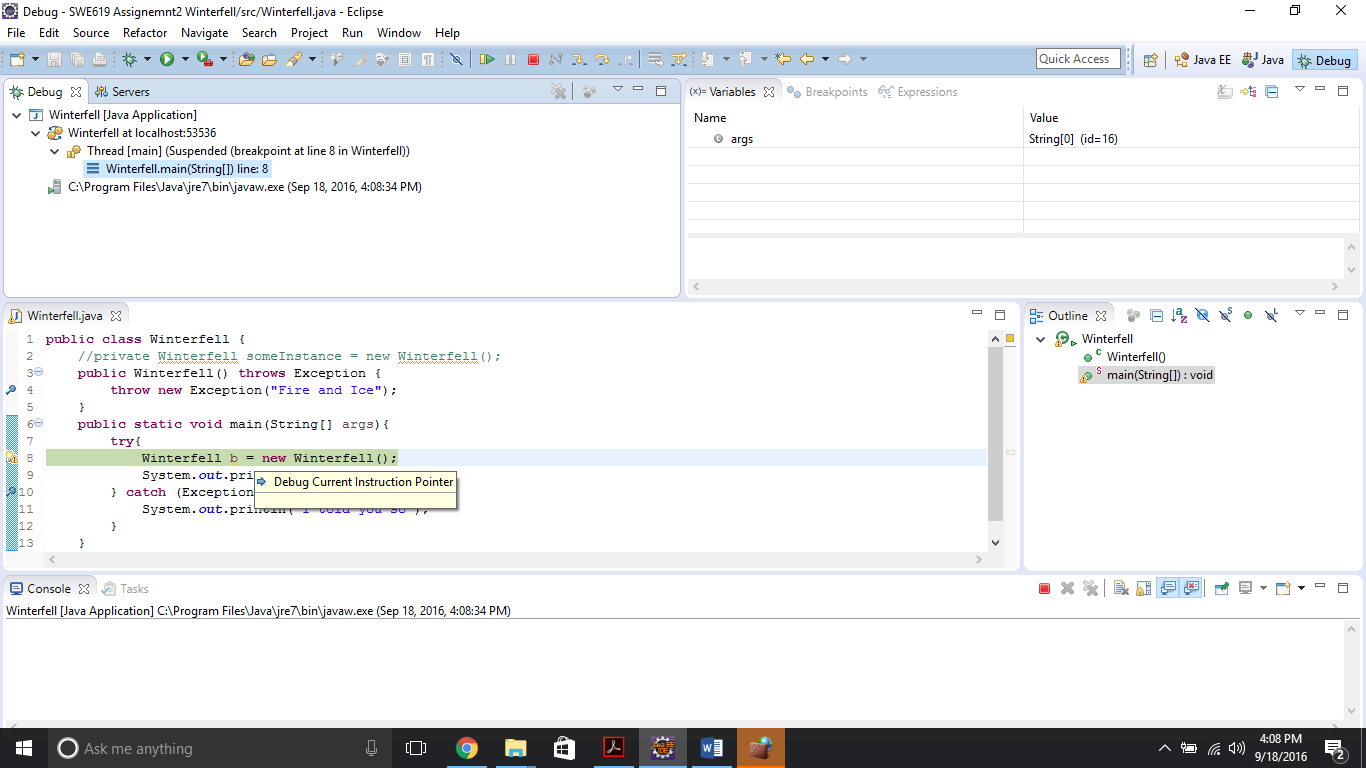
**Breakpoints:**

Line 4: **throw** **new** Exception("Fire and Ice");

Line 8: Winterfell b = **new** Winterfell();

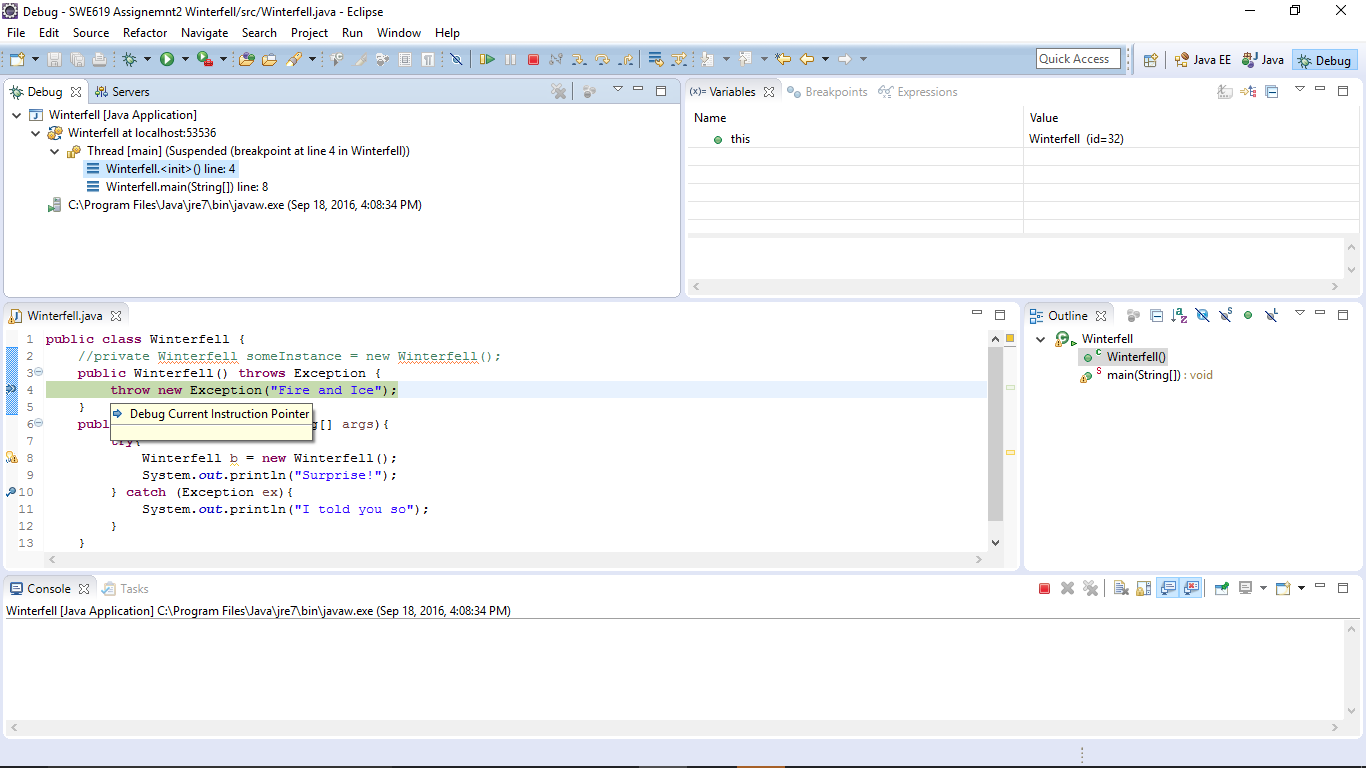
Line 10: System.***out***.println("I told you so");

When the debug starts, the control is passed to the main function of the class and thus breakpoint at line 8 is struck first **(as seen is screenshot below):**



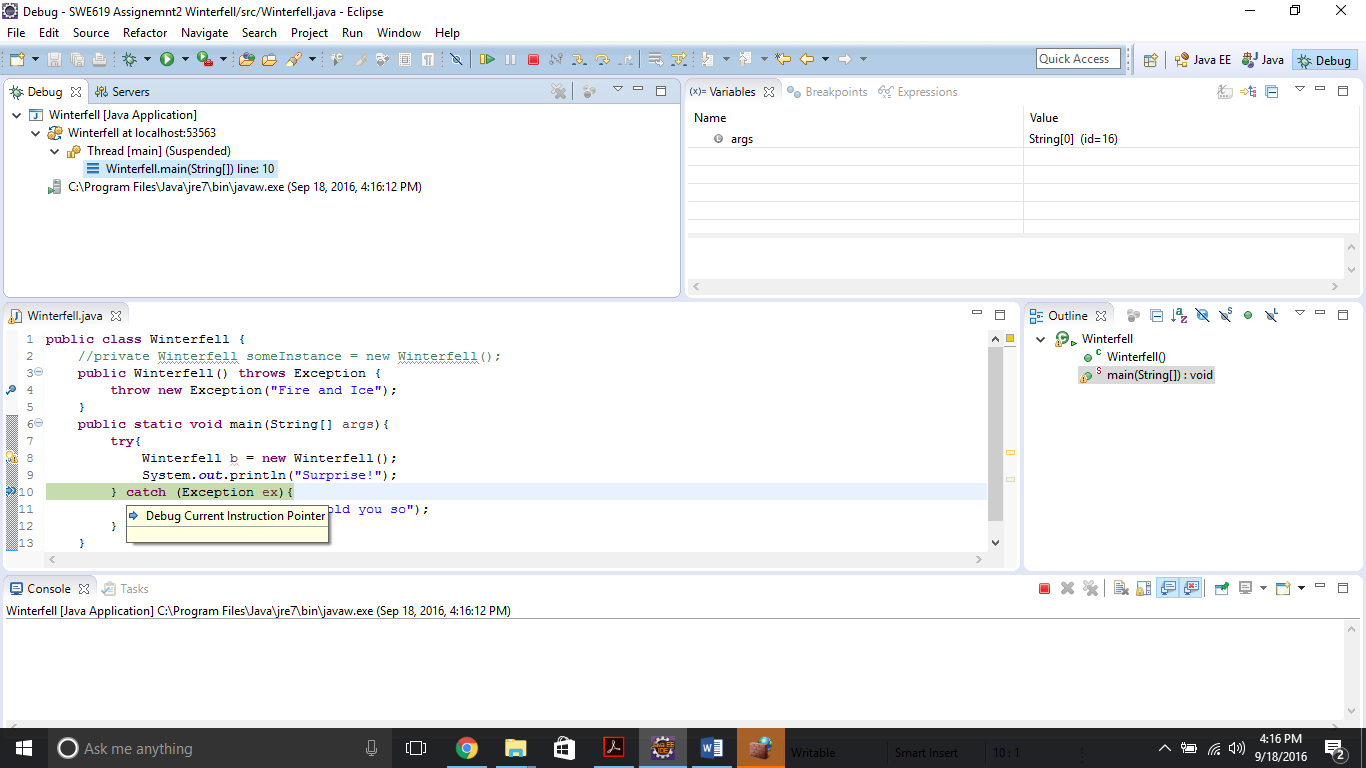
|  |  |
| --- | --- |
| **Variable value:** | |
| **Name** | **Value** |
| **args** | **String[0](id=16)**  **[]** |

**The control is then passed to the constructor of the class (as seen is screenshot below):**



|  |  |
| --- | --- |
| **Variable value:** | |
| **Name** | **Value** |
| **this** | **Winterfell (id=407)** Winterfell@7deb41d6 |

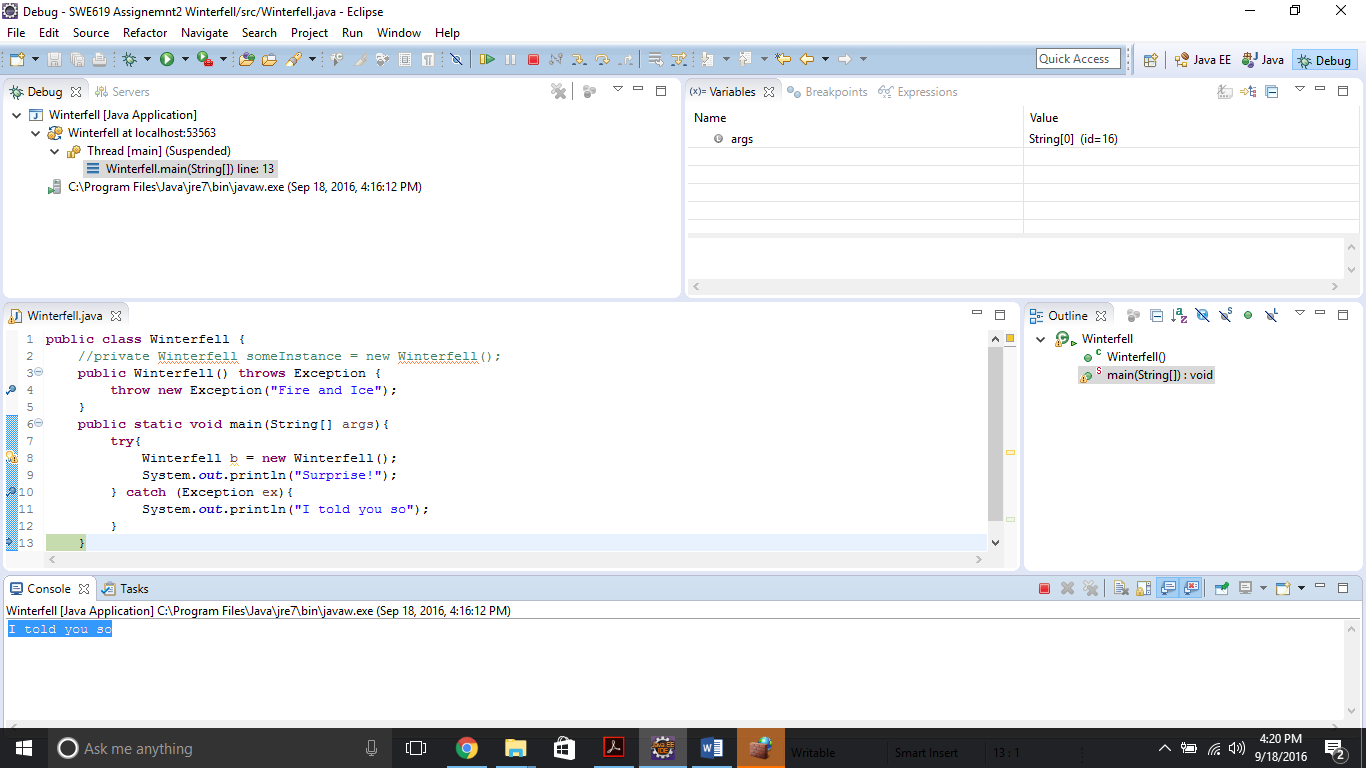
**The exception is caught and the next breakpoint is hit (as seen in screenshot below):**



|  |  |
| --- | --- |
| **Variable value:** | |
| **Name** | **Value** |
| **args** | **String[0] (id=16)** |
| **ex** | **Exception (id=409)**  java.lang.Exception: Fire and Ice |

**The output obtained:**

I told you so



**c) Explain the behavior. This is not easy. Hint: you will need to understand the Java runtime lifecycle for class. Take a look at the Java Language Specification 12.5.**

* When the code is compiled, the Winterfell.class a binary file is created.
* In the class, there is one more statement, **private Winterfell someInstance = new Winterfell();** This again creates an instance of the same class, which in turn creates the instance of the same class again. This execution goes into an infinite loop and creates a stack overflow error. So in order to execute/run the program successfully, we need to remove the line (rather I commented it).
* After the compilation, when execute (run) command is given, JVM extracts the details of the class (constants, references etc.) and then the method “**public static void main**” is executed.

**First scenario is according to the code discussed above:**

* According to our code, the execution first enters into “**try**” block and executes the statement “**Winterfell b = new Winterfell()**”;.
* As seen from the flow of breakpoints the execution goes as follows:
  + The constructor is implicitly called which has only one statement **throw new Exception(“Fire and Ice”);** and thus the exception is thrown**.**
  + The execution now enters the catch block (to catch the exception thrown) and the code inside the catch block is executed.

**System.*out*.println(“I told you so”);** is executed and thus we get output as ”**I told you so**”.

To consider another scenario i.e. with the following code where the statement **Winterfell b = new Winterfell();** is commented:

**public** **class** Winterfell {

//private Winterfell someInstance = new Winterfell();

**public** Winterfell() **throws** Exception {

**throw** **new** Exception(“Fire and Ice”);

}

**public** **static** **void** main(String[] args){

**try**{

//Winterfell b = new Winterfell();

System.***out***.println(“Surprise!”);

} **catch** (Exception ex){

System.***out***.println(“I told you so”);

}

}

}

* As the try block does not contain the statement **Winterfell b = new Winterfell();**, the class is not instantiated and thus so the constructor is not invoked and the exception statement is not executed.
* The control remains in the try block and code flow directly executes the statement **System.*out*.println("Surprise**!"); and the program simply gives the output, "Surprise!"