1. **Answer the following questions after executing the code in Fig 11.3**
   1. At which line number the “ **InputMismatchException** ” occur? (**1 point**)

스크린샷이(가) 표시된 사진

자동 생성된 설명

Answer: When 100, hello is entered, inputMismatchException is generated on line #27.

* 1. Why the “**InputMismatchException**” occurs? (1 point)

The input data type needs to be entered through scanner.nextInt(), but InputMismatchException occurs that the data types are not the same as each other because you have entered another data type, such as String data type.

* 1. At which line number the “**ArithmeticException”** occurs? (1 point)

스크린샷이(가) 표시된 사진

자동 생성된 설명

* + - * ArithmeticException occurs at Line#12, # 29, based on fig 11.3 when 100 and 0 are entered.
  1. Why the “**ArithmeticException**” occurs. (1 point)
     + - ArithmeticException occurs because the denominator cannot be zero when dividing a specific number, resulting in an exception in the mathematical calculation process.

1. **Answer the following questions about the program in Fig 11.3**
   1. Before line 34, **add** the following catch block and compile the program. Explain what happens and explain why this happens (including only explanation, **2 points**)

|  |
| --- |
| catch (**RuntimeException** re)  {  System.err.printf("%n Exception: %s%n", re) ;  } |

스크린샷이(가) 표시된 사진

자동 생성된 설명

* + - * Compile error occurred because two catch syntaxes associated with a specific try syntax handled an exception of the same type, as in the warning syntax.
  1. Replace the **second** catch block (Fig.11.3) by the following catch block. Then **enter** input similar values as shown in the example and execute the program. **Explain** what result you observed and explain why such type of results occur (including the **reason** and **screen shot of the result , 2 points**)

|  |
| --- |
| **Catch** (RuntimeException re)  {  System.err.printf("%nException: %s%n", re);  } |

스크린샷이(가) 표시된 사진

자동 생성된 설명

* + - * Normal case. Not error

스크린샷이(가) 표시된 사진

자동 생성된 설명

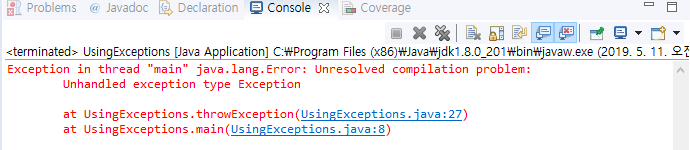
* + - * ArithmeticException is generated by entering 100 and 0. It is no different from ex1

스크린샷이(가) 표시된 사진

자동 생성된 설명

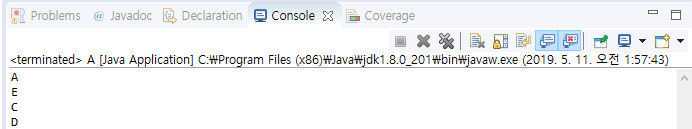
* + - * InputMismatchException occurs as in ex1. The reason is as above.

1. After deleting “**throws Exception**” at line 21(Fig 11.5), compile the program. Explain why a problem happens (**including only explanation, 2 points**)



If an exception is likely to occur in method throughException, the throughs Exception syntax should be placed behind the method throughException, as before deleting "throwsException", but the exception cannot be processed because the syntax was deleted. Thus, a compile error occurs, as shown in the screen shot above.

1. After **running**  the following code, explain the reason why you got this result (including your reason and screen shot of the result, **3 points**)



Print A first by ~

Try

{

System.out.println(“A”);

M();

System.out.println(“B”);

}

Then call method M.

**public** **static** **void** m()

{

System.***out***.println("E");

**if**(**true**)

**throw** **new** RuntimeException();

System.***out***.println("F");

}

* Print E

**catch**(Exception e)

{

System.***out***.println("C");

}

Then run through new RuntimeException(); because if(true) conditional statement is always true. Because RuntimeException occurred, output C of the main method.

Then run finally syntax because the end of the try syntax.

**finally**

{

System.***out***.println("D");

}

* So print D

Therefore print A, E, C, D.

|  |
| --- |
| // **A.java**  1 public class **A**  2. {  3. public static void **main**(String[] args)  4. {  5. try  6. {  7. System.out.println("A");  8. **m();**  9 . System.out.println("B");  10 }  11. catch(Exception e)  12 {  13. System.out.println("C");  14. }  15. finally  16. {  17. System.out.println("D");  18. }  19. }// end of main() method  21 public static void **m**()  22. {  23. System.out.println("E");  24. if(true)  25. **throw new RuntimeException**();  26. System.out.println("F");  27 }  28 }**// end of class** |

1. **When you compile the following program, an error occurs.**
   1. Explain **why** the error happens.
   2. In order to do **normal** operation, **add** the exception handling code using two ways. In your report, include screen shot of errors, explanation for the occurrence of the error, screen shot of the normal code (without error). Include also the source code of the normal code in the source file (**4 points**)

|  |
| --- |
| **import** java.io.\*;  **class** Demo1  {  **public** **static** FileInputStream **CreateFile**(String **fileName**)  {  FileInputStream **fis2** = **new** FileInputStream(**fileName**);  System.*out*.println("File input stream is created");  **return** **fis2**;  }  **public** **static** **void** main(String args[])  {  **FileInputStream fis1 = null;**  String **fileName** = "foo.bar";  System.*out*.println("File name is " **+** fileName);  Fis1 = ***CreateFile***(**fileName**);  System.*out*.println("End of the program");  }  **}** |