**Exception Hierarchy:**

**For all the JAVA classes – Object is the root for all**

**For all the exception and errors – Throwable is root**

**Throwable is class only not an Interface.**

**Throwable class contains 2 child classes.1) Exception and 2) Error**

**Most of the times exceptions are caused by our program only.Ex: FileNotFoundException – we are giving the instruction to read the file that’s why exception occurred.**

**Exceptions are recoverable.**

**Why exceptions are recoverable? If file not found in try block mean we will give the alternative solution in catch block to execute with local file.**

**Most of the times error will caused by our program which are caused by lack of system resources and these are non-recoverable.**

**Example:1) Some virus attached hence JVM will get damaged.2) Out of memory**

**Under exception there are several sub class exceptions are available.**

**1)Runtime Exception,2)IOException,3)Interrupted Exception,4)Remote Exception,5)SQL Exception**

**Under Runtime Exception there are several exceptions are available.**

**1)Arithmetic Exception,2)Null pointer Exception 3)Class caste Exception 4)Index out of bounds Exception---- Array Index out of bounds Exception, String Index out of bounds exception5)Illegal Argument Exception – Number format exception**

**Under IOException there are several exception classes are available.**

**1)EOFException(End of file Exception) and 2) File not found exception**

**Under Error there are several sub class exceptions are available.**

**1)VMError (1,Out of Memory Error and 2.Stack overflow Error)**

**2)linkageError (Verify Error)**

**3)Assertion Error**

**4)ExceptionInInitializer Error**

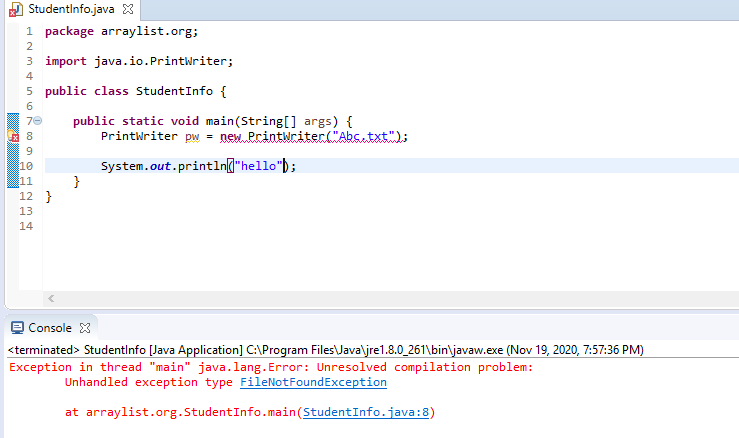
**Checked and Unchecked Exceptions:**

**Note: Every exception will be occurred at run time only and at compile time there will not be occurred any exception.**

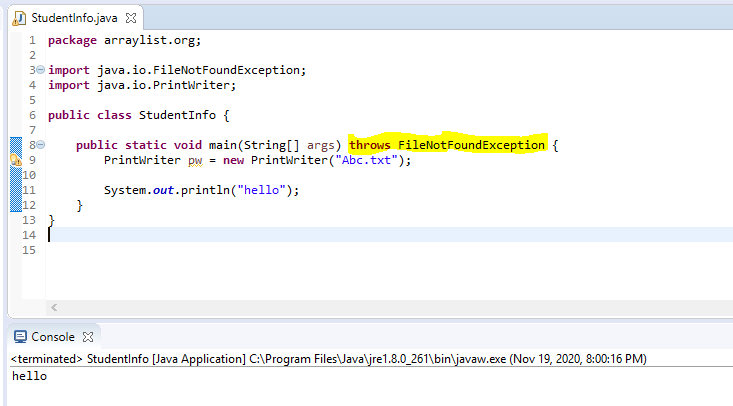
**At compile time maybe we will get synthetical errors.**

**Checked Exception: The exceptions which are checked by the compiler for smooth execution of the program at run time.**

**Checked exception should be handled by the programmer. (Ex: try catch)**



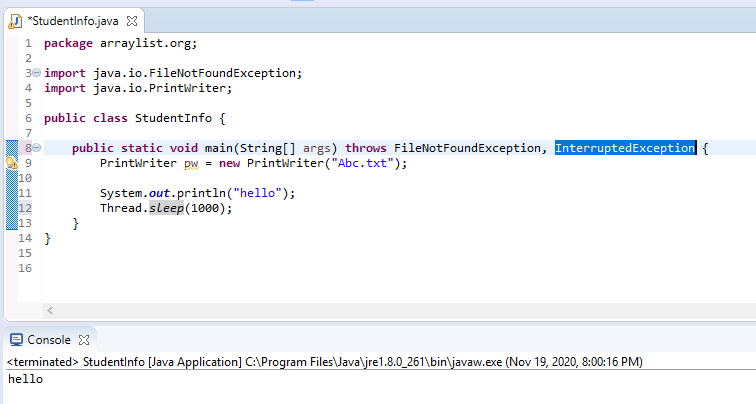
**In above program, Compiler will ask that if abc.txt file is not found mean what we need to do? If we didn’t provide any exception handling code mean we receive above error.**



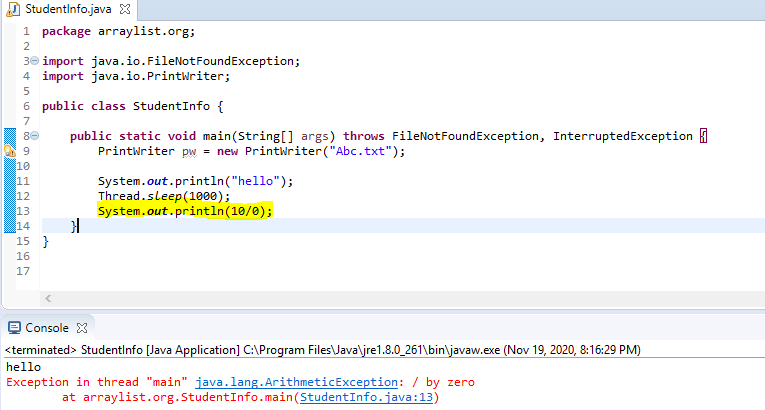
**If we pass the throws exception like above we won’t receive any compile time error.**

**Example 2:**

**If we give thread.sleep mean compiler will ask with the program like that did we handle any exception or not.If not pass the exception message and if pass mean no issue.**



**Unchecked exception: The exception which are not checked by the compiler and the coding also will get smooth execution but we receive the exception at run time**



**Which all are checked and unchecked?**

**All the errors and run time exception and their child classes are Unchecked exception.**

**Other than these like Throwable,interruptedException,IOException,SQLError,etc are checked exception.**

**Difference between fully checked and partially checked exceptions:**

**Exception and Throwable is only the partially checked exceptions since those some of the exceptions are unchecked (AE,NPE).**

**Interrupted and IOException,SQLException,RemoteException are checked exception since those exceptions are purely checked exception.**

**Practice:**

**IOException – Checked (Fully checked)**

**RuntimeException – Unchecked**

**InterruptedException- Checked (Fully checked)**

**Error – Unchecked(Fully unchecked)**

**Throwable – Checked (Partially checked)**

**ArithmeticException – Unchecked Exception**

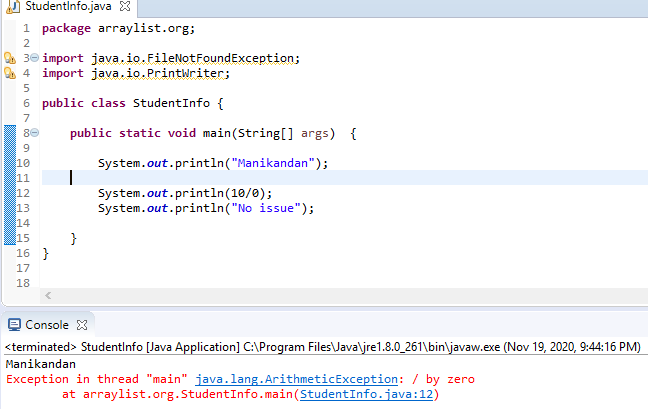
**Null pointer Exception – Unchecked Exception**

**Exception- Checked (Partially checked)**

**FileNotFoundException – Checked Exception**

**The usage of Exception handling:**

**Without Try catch block:**

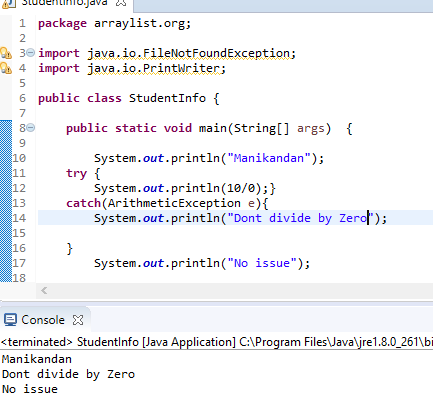


**Here, we are not provide any exception handling code for arithmetic exception.**

**Line 10 only is print and after receive AEException and then 13th line is not printed.Because this program is abnormal termination since 13th line is wastey entered.**

**To resolve the above issue, we have to use try catch block.**

**Try ----Risky code and catch--- handling code**



**The above program is normal termination (or) graceful termination since we didn’t miss anything.**