**Creation of custom Exceptions**

package Generics;  
  
public class GenericsInExceptions {  
 public static void main(String[] args) throws MyCustomException {  
 throw new MyCustomException("asf");  
 }  
}  
  
class MyCustomException extends Exception{  
  
 public <T> MyCustomException(T value){  
 super ("Exception related to value:"+ value.toString()+"of Type:"+value.getClass().getName());  
 }  
  
}

**C:\Users\Roystan\.jdks\openjdk-21.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.3\lib\idea\_rt.jar=56638:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.3\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\Roystan\IdeaProjects\JavaWorkspace\out\production\JavaWorkspace Generics.GenericsInExceptions**

**Exception in thread "main" Generics.MyCustomException: Exception related to value:asfof Type:java.lang.String**

**at Generics.GenericsInExceptions.main(GenericsInExceptions.java:5)**

**Process finished with exit code 1**

package Generics;  
  
public class GenericsInRunTimeExceptions {  
 public static void main(String[] args) {  
 throw new MyCustomException("asf");  
 }  
}  
  
class MyCustomException extends RuntimeException{  
  
 public <T> MyCustomException(T value){  
 super ("Exception related to value:"+ value.toString()+"of Type:"+value.getClass().getName());  
 }  
  
}

**C:\Users\Roystan\.jdks\openjdk-21.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.3\lib\idea\_rt.jar=56638:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.3\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\Roystan\IdeaProjects\JavaWorkspace\out\production\JavaWorkspace Generics.GenericsInExceptions**

**Exception in thread "main" Generics.MyCustomException: Exception related to value:asfof Type:java.lang.String**

**at Generics.GenericsInExceptions.main(GenericsInExceptions.java:5)**

**Process finished with exit code 1**

**Why Generic Exception class cannot be created**

* *Java does not support generic exception due to type erasure. Type erasure means that generic type information is removed during runtime. Since exceptions are closely tied to runtime operations(like catching them in try-catch blocks),having generic exceptions wouldn’t work as expected*
* *Catch blocks in java relies on the exact type of exception class at runtime to determine which block to execute. Since generics are erased at runtime,the type parameter of a generic exception class is not available, making it* ***impossible to disguise between generic exception in a catch block***
* *Since type information is erased during runtime cannot determine whether te MyException<String> matches MyException<Integer>*