[Solved] java.lang.ClassCastException using Generics in Java

total-qa August 24, 2018 No Comments

[Solved] java.lang.ClassCastException using Generics in Java

Generally when ever we write the **Java** Logic its better to know the issues during the compile time rather than at run time. If we do not use **Generics** we might face **java.lang.ClassCastException** when trying to add an Integer Object instead of String Object and try to retrieve the value and perform an operation on it will cause this **exception**. How do we overcome this using **Generics** we will learn from this tutorial.

Lets implement the Logic without using the Generics:

```
import java.util.ArrayList;
import java.util.List;
public class ClassCastExceptionExample {

    public static void main(String[] args) {
        String str1 = new String("RC");
        String str2 = new String("Selenium");
        List  strList = new ArrayList();//Without using Generics
        strList.add(str1);
        strList.add(str2);
        strList.add(2);//Adding an integer instead of String

        String str = (String) strList.get(2);
        str.substring(0, 2);//Throws ClassCastException of performing an opera

    }
}
```

Executing this Program returns following output in the Console:

```
Exception in thread “main” java.lang.ClassCastException: java.lang.Integer cannot be cast to java.lang.String
at org.h2k.javaprograms.ClassCastExceptionExample.main(ClassCastExceptionExample.java:15)
```

Lets implement the Logic using the Generics:

In the above program change the Line No:8 as follows:

```
List<String> strList = new ArrayList<String>();
```

```
1  import java.util.ArrayList;
2  import java.util.List;
3  public class ClassCastExceptionExample {
4
5      public static void main(String[] args) {
6          String str1 = new String("RC");
7          String str2 = new String("Selenium");
8          List<String>  strList = new ArrayList<String>();
9          strList.add(str1);
10         strList.add(str2);
11         strList.add(2);//Compile Time Error at Line number 13
12
13         String str = (String) strList.get(2);
14         str.substring(0, 2);//Throws ClassCastException of performing an opera
15
16     }
17 }
```

Conclusion:

Therefore implementing Generics avoids **ClassCastException** During run time. Any issues we can caught during Compile Time.