package Generics;  
  
public class Generics {  
 public static void main(String[] args) {  
 Box asd=new Box(12);  
 String k=(String) asd.getValue();  
 }  
}  
  
class Box {  
 private Object value;  
  
 Box(Object value){  
 this.value=value;  
 }  
  
 public Object getValue(){  
 return this.value;  
 }  
}

Output

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=61172: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.Generics

Exception in thread "main" java.lang.ClassCastException: class java.lang.Integer cannot be cast to class java.lang.String (java.lang.Integer and java.lang.String are in module java.base of loader 'bootstrap')

at Generics.Generics.main(Generics.java:6)

Process finished with exit code 1

**Separate Box class should have created for each data type which could have created duplication of code. To avoid it we have to use generics as below**

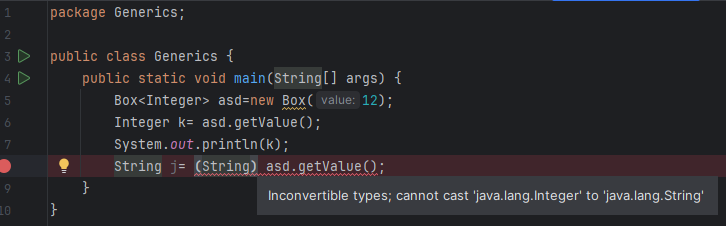
package Generics;  
  
public class Generics {  
 public static void main(String[] args) {  
 Box<Integer> asd=new Box(12);  
 Integer k= asd.getValue();  
 System.*out*.println(k);  
  
 }  
}  
  
class Box <T>{  
 private T value;  
  
 Box(T value){  
 this.value=value;  
 }  
  
 public T getValue(){  
 return this.value;  
 }  
}

**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=59197: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.Generics**

**12**

**Process finished with exit code 0**

**Compile time error is displaced if we cast to other String for above example**

****

**EXAMPLE 2**

package Generics;  
  
public class KeyValueGenericsExample {  
 public static void main(String[] args) {  
 Pair<Integer,String> pair=new Pair<>(1,"Roystan");  
 Integer key=pair.getKey();  
 String value= pair.getValue();  
 System.*out*.println("key: "+key+" value: "+value);  
  
 }  
  
}  
  
class Pair <K,V>{  
  
 private K key;  
 private V value;  
  
 Pair(K key, V value){  
 this.key=key;  
 this.value=value;  
 }  
  
 public K getKey(){  
 return key;  
 }  
  
 public V getValue(){  
 return value;  
 }  
  
}

**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=59276: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.KeyValueGenericsExample**

**key: 1 value: Roystan**

**Process finished with exit code 0**