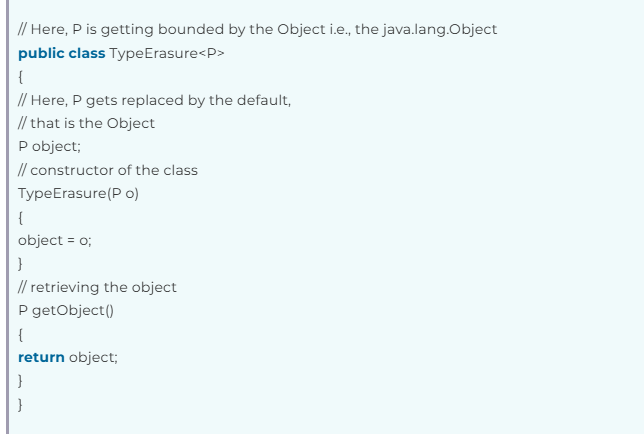
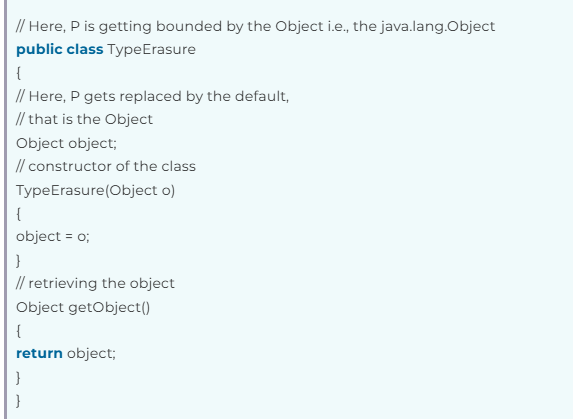
***Type Erasure in Java***

* *During Run Time -> Replacing all parameters type in the generic types with the bounds or Object when the type of the parameters is not bounded. The generated bytecode, hence, only contains the ordinary methods, interfaces, and classes*.
* *Inserting the type casts if required in order to support the type safety.*
* *Generating the bridge methods for preserving polymorphism in generic types that are extended.*

***Example***

******

***Typed Erasure for above Example***

******

**Example 1**

package Generics;  
  
import java.util.Arrays;  
import java.util.List;  
  
public class Generics {  
 public static void main(String[] args) {  
 Box<String> asd=new Box("Stringas");  
 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;  
 }  
}

**after type erasure**

package Generics;  
  
public class TypeErasureDemo1 {  
 public static void main(String[] args) {  
 Box asd=new Box("Stringas");  
 String k= (String) asd.getValue();  
 System.*out*.println(k);  
 }  
  
}  
  
class Box {  
 private Object value;  
  
 Box(Object value){  
 this.value=value;  
 }  
  
 public Object getValue(){  
 return this.value;  
 }  
}

**Example 2**

package Generics;  
  
import java.util.Arrays;  
import java.util.List;  
  
public class Generics {  
 public static void main(String[] args) {  
 Box<Integer> asd=new Box(12);  
 String 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;  
 }  
}

**after type erasure**

package Generics;  
  
public class TypeErasureDemo1 {  
 public static void main(String[] args) {  
 Box asd= new Box(12);  
 Integer k= (Integer) asd.getValue();  
 System.*out*.println(k);  
 }  
  
}  
  
class Box {  
 private Number value;  
  
 Box(Number value){  
 this.value=value;  
 }  
  
 public Number getValue(){  
 return this.value;  
 }  
}