**UNSAFE ASSIGNMENT IN JAVA WITHOUT GENERICS:**

|  |  |
| --- | --- |
| **package** com.company;  **import** org.omg.PortableInterceptor.SYSTEM\_EXCEPTION;  **public class** Main {   **public static void** main(String[] args) {  nonGenerics nn = **new** nonGenerics(8);  nn.showType(); *//type of the object is : java.lang.Integer* **int** v = (Integer) nn.getOb();  System.***out***.println(**"value is : "** + v); *// 8* System.***out***.println();   nonGenerics strnn = **new** nonGenerics(**"this is java Generics"**);  strnn.showType(); *//java.lang.String* String str = (String) strnn.getOb();  System.***out***.println(**"value is : "** + str);   System.***out***.println();   nn = str; *//unsafe type conversation* v = (Integer) nn.getOb();  } } | **package** com.company;  **public class** nonGenerics {  Object **ob**;   **public** nonGenerics(Object ob) {  **this**.**ob** = ob;  }   **public** Object getOb() {  **return ob**;  }   **public void** showType(){  System+.***out***.println(**"type of the object is : "** + **ob**.getClass().getName());  } } |
| **type of the object is : java.lang.Integer**  **value is : 8**  **type of the object is : java.lang.String**  **value is : this is java Generics** | |

**WITH GENERICS:**

|  |  |
| --- | --- |
| **package** com.company;  **import** org.omg.PortableInterceptor.SYSTEM\_EXCEPTION;  **public class** Main {   **public static void** main(String[] args) {  Generics<Integer> generics = **new** Generics<Integer>(8);  generics.showType();  **int** v = generics.getOb();  System.***out***.println(**"the vale : "** + v);   System.***out***.println();   Generics<String > generics1 = **new** Generics<String>(**"i love to do java"**);  generics1.showType();  String v1 = generics1.getOb();  System.***out***.println(**"value is : "** + v1);  } } | **package** com.company;  **public class** Generics<T> {  T **ob**;   **public** Generics(T ob) {  **this**.**ob** = ob;  }   **public** T getOb() {  **return ob**;  }   **public void** showType(){  System.***out***.println(**"type is : "** + **ob**.getClass().getName());  } } |
| **type is : java.lang.Integer**  **the vale : 8**  **type is : java.lang.String**  **value is : i love to do java** | |

**WITH GENERICS with 2 parameters : that has 2 parameters**

|  |  |
| --- | --- |
| **package** com.company;  **import** org.omg.PortableInterceptor.SYSTEM\_EXCEPTION;  **public class** Main {   **public static void** main(String[] args) {  Generics<Integer, String> generics = **new** Generics<Integer, String>(8, **"python is shit"**);  generics.showType();  generics.showType1();  **int** v = generics.getOb();  String vStr = generics.getOb1();  System.***out***.println(**"the vale : "** + v);  System.***out***.println(**"the vale : "** + vStr);   System.***out***.println();   Generics<String, Double> generics1 = **new** Generics<String, Double>(**"i love to do java"**, 9.8);  generics1.showType();  generics1.showType1();  String v1 = generics1.getOb();  Double v1D = generics1.getOb1();  System.***out***.println(**"value is : "** + v1);  System.***out***.println(**"value is : "** + v1D);  } } | **package** com.company;  **public class** Generics<T, U> {  T **ob**;  U **ob1**;   **public** Generics(T ob, U ob1) {  **this**.**ob** = ob;  **this**.**ob1** = ob1;  }   **public** T getOb() {  **return ob**;  }   **public** U getOb1(){  **return ob1**;  }   **public void** showType(){  System.***out***.println(**"type is : "** + **ob**.getClass().getName());  }   **public void** showType1(){  System.***out***.println(**"type is : "** + **ob1**.getClass().getName());  } } |
| **type is : java.lang.Integer**  **type is : java.lang.String**  **the vale : 8**  **the vale : python is shit**  **type is : java.lang.String**  **type is : java.lang.Double**  **value is : i love to do java**  **value is : 9.8** | |