**Array**

Interesting facts about Array assignment in Java

Prerequisite :[Arrays in Java](http://www.geeksforgeeks.org/arrays-in-java/)

While working with arrays we have to do 3 tasks namely declaration, creation, initialization or Assignment.  
Declaration of array :

int[] arr;

Creation of array :

// Here we create an array of size 3

int[] arr = new int[3];

Initialization of array :

arr[0] = 1;

arr[1] = 2;

arr[3] = 3;

int intArray[]; // declaring array

intArray = new int[20]; // allocating memory to array

**Some important facts while assigning elements to the array:**

1. **For primitive data types :** In case of primitive type arrays, as array elements we can provide any type which can be **implicitly promoted to the declared type array**. Apart from that, if we are trying to use any other data-types then we will get compile-time error saying possible loss of precision.

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on int type array  public class Test {  public static void main(String[] args)      {          int[] arr = new int[3];          arr[0] = 1;          arr[1] = 'a';          byte b = 10;          arr[2] = b;          System.out.println(arr[0] + arr[1] + arr[2]);      }  } |

1. Run on IDE
2. Output:
3. 108

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on int type array  public class Test {  public static void main(String[] args)      {          int[] arr = new int[3];            // Assigning long value to int type.          arr[0] = 10l;          arr[1] = 'a';          byte b = 10;          arr[2] = b;            System.out.println(arr[0] + arr[1] + arr[2]);      }  } |

1. Run on IDE
2. Output:
3. possible loss of precision.

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on char type array  public class Test {  public static void main(String[] args)      {          char[][] arr = new char[2][2];            // Assigning long value to int type.          arr[0][0] = 10l;          arr[0][1] = 'a';          char b = 10;          arr[1][0] = b;            // Assigning double value to char type          arr[1][1] = 10.6;      }  } |

1. Run on IDE
2. Output:
3. error: incompatible types: possible lossy conversion from long to char
4. error: incompatible types: possible lossy conversion from double to char
5. **Object type Arrays :**If we are creating object type arrays then the elements of that arrays can be either declared type objects or it can be child class object.

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on Number type array  public class Test {  public static void main(String[] args)      {          Number[] num = new Number[2];          num[0] = new Integer(10);          num[1] = new Double(20.5);          System.out.println(num[0]);          System.out.println(num[1]);      }  } |

1. Run on IDE
2. Output:
3. 10
4. 20.5

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on Number type array  public class Test {  public static void main(String[] args)      {          Number[] num = new Number[3];          num[0] = new Integer(10);          num[1] = new Double(20.5);            // Here String is not the child class of Number class.          num[2] = new String(“GFG”);      }  } |

1. Run on IDE
2. Output:
3. Compile-time error(incompatible types)

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on Number type array  public class Test {  public static void main(String[] args)      {          Number[][] arr = new Number[2][2];          arr[0][0] = 10l;            // Assigning char to Number type array          arr[0][1] = 'a';          byte b = 10;          arr[1][0] = b;            // Assigning String to Number type array          arr[1][1] = "GEEKS";      }  } |

1. Run on IDE
2. Output:
3. error: incompatible types: char cannot be converted to Number
4. error: incompatible types: String cannot be converted to Number
5. **Interface type array :** For interface type array, we can assign elements as its implementation class objects.

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on Interface type array  public class Test {  public static void main(String[] args)      {          Runnable[] run = new Runnable[2];            // Thread class implements Runnable interface.          run[0] = new Thread();          run[1] = new Thread();      }  } |

1. Run on IDE

|  |
| --- |
| // Java program to illustrate the concept of array  // element assignments on Interface type array  public class Test {  public static void main(String[] args)      {          Runnable[] run = new Runnable[2];          run[0] = new Thread();            // String class does not implements Runnable interface.          run[1] = new String(“GFG”);      }  } |

1. Run on IDE
2. Output:
3. Compile-time error(Incompatible types)