Character strings represents the string class and all string literals in Java programs are implemented as instances of this class. Strings are constant and once they have been created, their values cannot be changed String buffers support mutable strings and string objects are immutable that can be shared. Therefore, String str = “abc”; is equivalent to char data [] = {‘a’, ‘b’, ‘c’}; String str = new String (data);. Strings can be used in several different ways including System.out.printIn(“abc”);, String cde = “cde”;, System.out.printIn (“abc” + cde);, String c = “abc”.substring(2,3);, and String d = cde.substring(1, 2);. The class string includes methods for examining individual characters of the sequence, comparing strings, searching strings, extracting substrings, and creating a copy of a string with all the carachters translated to uppercase or to lowercase.

All methods include static methods, instance methods, concrete methods, and deprecated methods. The char type is used in a method of charAt (int index) to return the char value at the specified index. An index ranges from 0 to length () – 1 and the first char value of the sequence is at index 0, the next at index 1 and so on for array indexing. The int type can use the methods codePointAt (int index) to return the character at the specified index, codePointBefore (int index) to return the character before the specified index, codePointCount (int beginIndex, int endIndex) to return the number of Unicode code points in the specified text range of the string, compareTo (String anotherString) to compare two strings lexicographically, and compareToIgnoreCase (string str) to compare two strings lexicographically and ignoring case differences. The Boolean type can use the methods contains (CharSequence s) to return true if and only if the string contains the specified sequence of character values, contentEquals (CharSequence cs) to compare the string to the specified CharSequence, and contentEquals (StringBuffer sb) to compare the string to the specified StringBuffer. The static String utilizes different forms of valueOf for different data types to return the string representation of the specific data type argument. Additionally, the string type can use toLowerCase () to convert all of the characters to lower case using the rules of the default locale, toLowerCase (Locale locale) to convert all of the characters to lower case using the rules of the given locale, the same for uppercase (toUpperCase () and toUpperCase (Locale locale)) and toString () where the object, which is already a string, is itself returned.

Public class Arrays contains various methods for manipulating arrays such as sorting and searching containing a static factory that allows arrays to be viewed as lists. The methods for arrays all throw a NullPointerException if the specified array reference is null except where otherwise noted. The documentation for the methods includes brief descriptions of the implementations and they should be regarded as implementation notes rather than parts of the specification. The implementers should feel free to substitute other algorithms so long as the specification itself is adhered to. All of the methods for arrays are static.

The sort method sorts the specified array into a particular order. This can be done in many different ways such as ascending numerically, the natural ordering of its elements, and general ascending order. An example of this would be: public static void short (long [] a). The binarySearch method searches the specified array for a specified value using the binary search algorithm. An example of this would be:

Public static int binarySearch (float [] a,

Float key)

The equals method returns true if two specified arrays of longs, shorts, ints, chars, bytes, Booleans, doubles, floats, and objects are equal to one another. An example of this would be:

Public static Boolean equals (double [] a,

Double [] a2)

The fill method assigns a specific value to each element of the specified array of longs, ints, shorts, chars, bytes, Booleans, doubles, floats, and objects. An example of this would be:

Public static void fill (Object [] a,

Int fromIndex,

Int toIndex,

Object val)

The copyOf method copies the specified array at a specified length with nulls or zeros for two identical values. An example of this would be:

Public static <T> T[] copyOf (T[] original,

Int newLength)

<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#charAt(int)>

<https://docs.oracle.com/javase/9/docs/api/java/lang/String.html>

<https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>