**Java Notes:**

1. **OOPS concepts:**

There are four object-oriented programming principles. We would be discussing each of the principle in details .

* 1. **Encapsulation:**

Encapsulation is one of fundamental principle of object-oriented programming. Encapsulation is hiding/protecting data from getting corrupted inadvertent/advertently.

Encapsulation is done by making member variable as private and allowing the data accessible through method/functions this leads to encapsulation of data. Methods/function can help in checking the data for validity hence data is protected.

In java encapsulation achieved by proving getter and setter method and making variables as private.

With encapsulation we can make the data as read only by only providing getter methods.

Here is example of encapsulation in java:

“public class EncapExample{

private int value;

private getValue(){

return value;

}

private int setValue(int v)

if(v>0)

value=v;

}”

* 1. **Inheritance**
  2. **Abstraction**
  3. **Polymorphism**

1. **Java basic constructs**
   1. **Data Types**
   2. **Loops**
2. **String Handling**

Here are list of important method in the string class and their brief description. Please refer to java documentation for full list of methods.

1.[charAt](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#charAt-int-)(int index): Returns the char value at the specified index.

2. [compareTo](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#compareTo-java.lang.String-)([String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) anotherString):Compares two strings lexicographically(alphabetically)

3. [compareToIgnoreCase](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#compareToIgnoreCase-java.lang.String-)([String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) str)Compares two strings lexicographically, ignoring case differences.

4. [concat](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#concat-java.lang.String-)([String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) str): Concatenates the specified string to the end of this string.

5. [contains](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#contains-java.lang.CharSequence-)([CharSequence](https://docs.oracle.com/javase/8/docs/api/java/lang/CharSequence.html" \o "interface in java.lang) s):Returns true if and only if this string contains the specified sequence of char values.

6. [getBytes](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#getBytes--)():Encodes this String into a sequence of bytes using the platform's default charset, storing the result into a new byte array.

7.[indexOf](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#indexOf-int-)(int ch):Returns the index within this string of the first occurrence of the specified character.

8.[indexOf](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#indexOf-int-int-)(int ch, int fromIndex):Returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.

9.[indexOf](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#indexOf-java.lang.String-int-)([String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) str, int fromIndex):Returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

10.[isEmpty](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#isEmpty--)(): Returns true if, and only if, [length()](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#length--) is 0.

11.[length](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#length--)():Returns the length of this string

12. [split](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#split-java.lang.String-)([String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) regex):Splits this string around matches of the given [regular expression](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html#sum).

13. [split](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#split-java.lang.String-int-)([String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) regex, int limit): Splits this string around matches of the given [regular expression](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html#sum).

14.[substring](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#substring-int-)(int beginIndex):Returns a string that is a substring of this string.

15. [substring](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#substring-int-int-)(int beginIndex, int endIndex):Returns a string that is a substring of this string.

16. [toCharArray](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#toCharArray--)(): Converts this string to a new character array.

17. [toLowerCase](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#toLowerCase--)(): Converts all of the characters in this String to lower case using the rules of the default locale .

18. [toUpperCase](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#toUpperCase--)():Converts all of the characters in this String to upper case using the rules of the default locale.

19. [trim](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#trim--)(): Returns a string whose value is this string, with any leading and trailing whitespace removed.

1. **Exception Handling**
2. **Multithreading**
3. **Collection Framework**
4. **JDBC**
5. **Servlet**
6. **JSP**