String Builder

Click to Continue

StringBuilder class, is alternative for StringBuffer.

- StringBuffer is synchronized. StringBuilder is not synchronized. So StringBuilder is faster than StringBuffer.

Try It Out – StringBuilder API

Click to Continue

Develop a program "StringBuilderDemo.java" with a main method. And perform the following logic,

1. Append two Strings "Good" & "Morning" without using "+" operator. Store it in a StringBuilder variable S1 and print it. Result: "Good Morning"
2. Insert a string "$Jack" in the String S1 after "Morning". Store it in a StringBuilder variable S2 and print it. Result: "Good Morning$Jack"
3. Replace $ with space in S2. Store it in a StringBuilder variable S3 and print it. Result: "Good Morning Jack".

```
public class StringBuilderDemo {

    public static void main(String[] args) {
        StringBuilder s1 = new StringBuilder("Good").append("Morning");
        System.out.println(s1);
        StringBuilder s2 = new StringBuilder(s1.insert(11, "$Jack"));
        System.out.println(s2);
        StringBuilder s3 = new StringBuilder(s2.replace(11, 12, " "));
        System.out.println(s3);

    }
```

StringTokenizer

Click to Continue

StringTokenizer is a class used to break the String to tokens based on a delimiter.

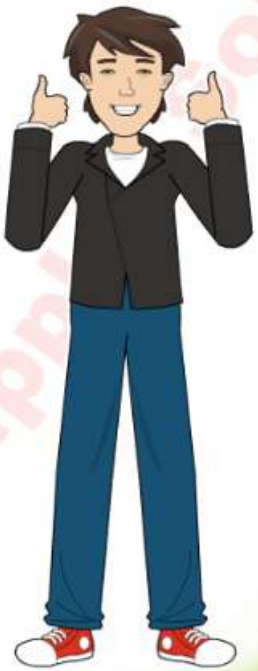
Illustration: Apple, Oranges, Guava, Pineapple – This string can be split based on the delimiter ","

The given string will be split into individual tokens,

Token 1- Apple ***Token 2-*** Oranges ***Token 3-*** Guava ***Token 4-*** Pineapple

hashCode()

- This method is present in class ***java.lang.Object***.
- This method returns the hash code value of a object .
- ***Hash code*** is an unique number allocated to an object by the JVM.
- You can override this method and develop your own logic of generating hash code.



equals() method

Click to Continue

equals() is a method present in class ***java.lang.Object***. This method is used for comparing the equality of two objects.

equals():

- ***equals()*** returns true if the two objects are equal.
- By default this compares the hash code of the objects.
- ***equals()*** method can be overridden with a logic which needs to be implemented for checking equality.

Illustration: Assume we have a Student object with instance variables, StudentId.

If we compare one student object with other using equals() this will return false as this by default compares hashcode.

The developer can override the equals method and compare the employee Id for checking equivalence.

Try it out – Equals and Hash Code

Click to Continue

Develop a class Student.java with studentId int as instance variable, perform the following logic,

NOTE: Refer to the java documentation java.lang.Object to find the method signature of the below two methods,

- Develop the *Equals* method and implement a logic to compare the id of student if same they should return a true else return false.
- *hashCode* – Should return the student id as hash code.

Develop the program given in next slide and lanalyze the results.

Try it Out - Solution

Click to Continue

```
package com.demo.string;

public class StudentVO {
    private int studentId;

    public int getStudentId() {
        return studentId;
    }

    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }

    public boolean equals(Object obj) {
        StudentVO vo = (StudentVO) obj;
        return (this.studentId == vo.studentId);
    }

    public int hashCode()
    {
        return studentId;
    }

}
```

```
package com.demo.string;

public class StudentManager {

    public static void main (String args[])
    {
        StudentVO vo1 = new StudentVO();
        vo1.setStudentId(10);


        StudentVO vo2 = new StudentVO();
        vo2.setStudentId(10);

        System.out.println("Message 1-->"+vo1.equals(vo2));
        System.out.println("Message 2-->"+vo1.hashCode());
        System.out.println("Message 3-->"+vo2.hashCode());
        System.out.println("Message 4-->"+(vo2==vo1));

        StudentVO vo3 = new StudentVO();
        vo1.setStudentId(100);


        StudentVO vo4 = new StudentVO();
        vo2.setStudentId(101);

        System.out.println("Message 5-->"+vo3.equals(vo4));
        System.out.println("Message 6-->"+vo3.hashCode());
        System.out.println("Message 7-->"+vo4.hashCode());
        System.out.println("Message 4-->"+(vo3==vo4));
    }
}
```

Please run the program and check what is the output.