# **Automation Strategy**

Automation Strategy is the document that is created for the team to guide them from technical front of Automation

1. Scope
2. Kind of Automation testing (performance, UI, Security testing)
3. fwfvg
4. Features to be automated and not to automated
5. Identify automation and non-automation candidates
6. Automation framework Architecture
7. Automation Environment
8. Test case design coding guidelines
9. Automation framework and Automation script guidelines
10. Metrics

**Framework considerations**

- Ease of Use

- Object Oriented Framework

- Application-Independent

- Easy Maintenance

- Pluggable framework components

- Reusable components

- Loosely coupled & Independent components

- Encapsulate the Framework implementation team from framework complexities

- Isolate the test data and object repository from the Framework Code

-Modularity

-Reusability

- Error handling

- Detailed Audit logs & Reporting

- CI Integration support

- Framework Fault Tolerance Support

**AUTOMATION metrics**

Automation coverage

Automation Progress

**Metrics categories:**Most software testing metrics (including the ones presented here) fall into one of three categories:

* **Coverage:**meaningful parameters for measuring test scope and success.
* **Progress:** parameters that help identify test progress to be matched against success criteria. Progress metrics are collected iteratively over time. They can be used to graph the process itself (e.g., time to fix defects, time to test, etc.).

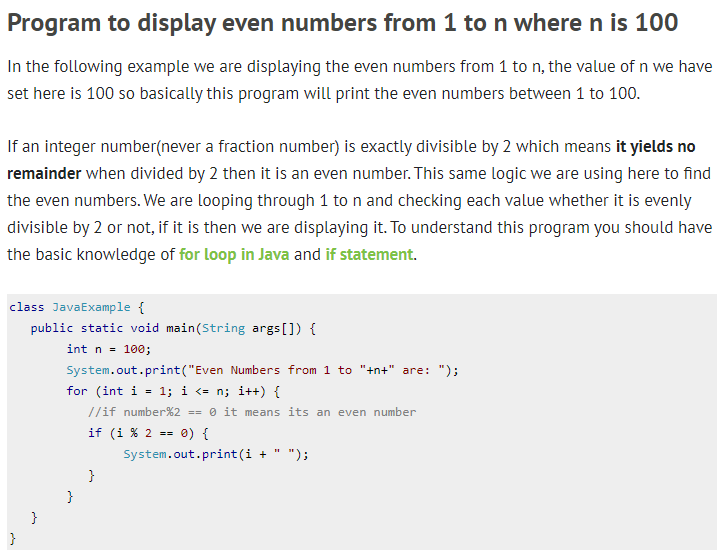
Hash map Internal Implementation

HashMap in Java stores both key and value object, in bucket, as an object of Entry class which implements this nested interface Map.Entry. ... When get() method is used to retrieve value, again key object is used to calculate a hash which is used then to find a bucket where that particular key is stored.

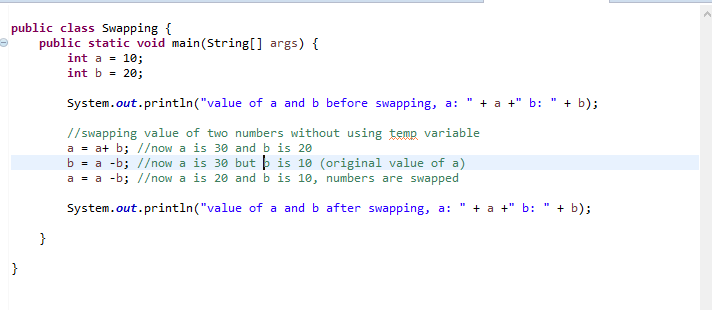
# Finonacci Series

# 

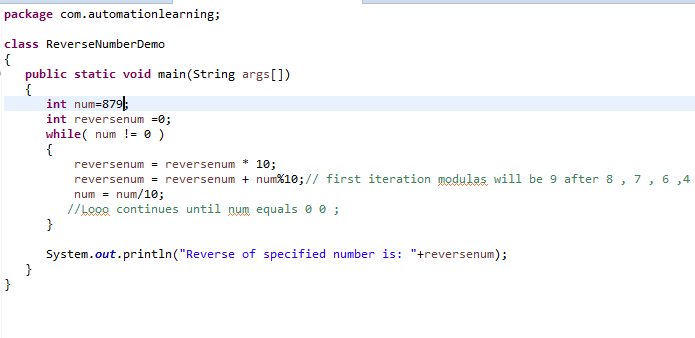
# Even number s



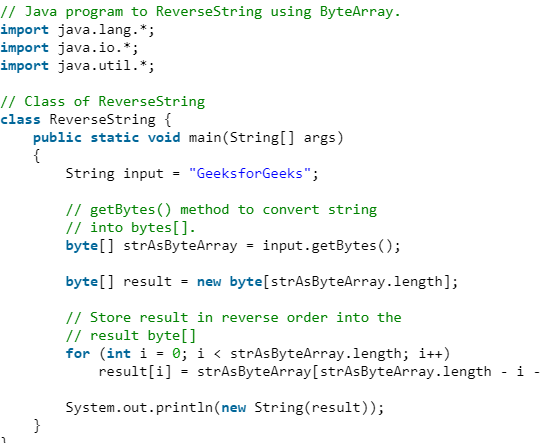
# ODD Numbers



**reverse a number?**

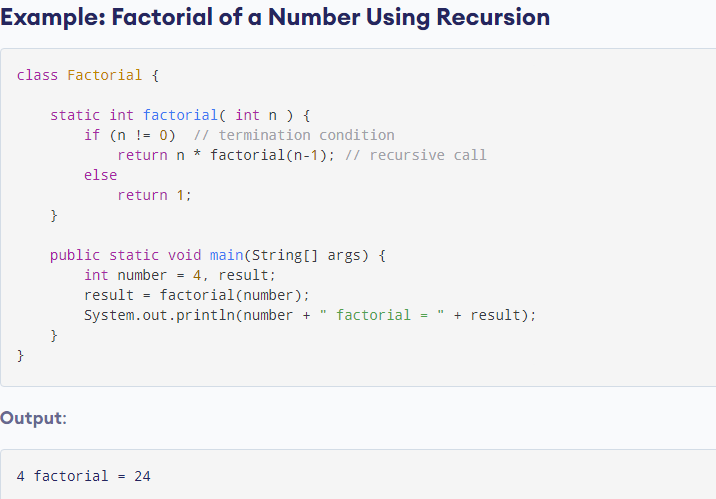


# Reverse a String



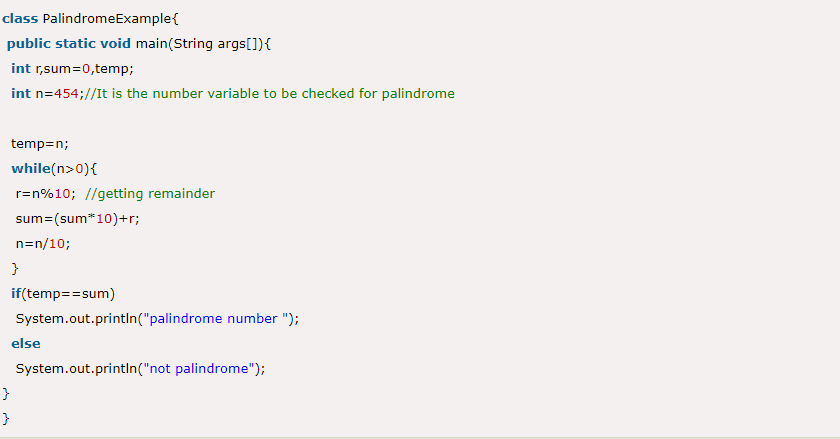
Contents

# **Recursion Factorial**

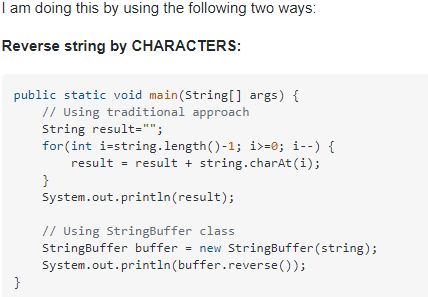


# **Palindrome**

**Number Palindrome**



**String Palindrome**



# **Program to swap two numbers without using the third variable**

This program is to swap/exchange two numbers without using the third number in the way as given below:

**Example:** Suppose, there are two numbers 25 and 23.

Let

**X= 25** (First number), **Y= 23** (second number)

Swapping Logic:

X = X + Y = 25 +23 = 48

Y = X - Y = 48 - 23 = 25

X = X -Y = 48 - 25 = 23

and the numbers are swapped as X =23 and Y =25.

# **Count occurrences of a word in string**

Graphical user interface, text, application, email

Description automatically generated