# Dhole Patil College of Engineering, Pune

Savitribai Phule Pune University (SPPU)
Fourth Year of Computer Engineering (2019 Course)

410246: Laboratory Practice III

Subject Teacher: - Prof. Suchitra Deokate(DAA)
Prof. Suchitra Deokate(ML)
Prof. Archana Priyadarshani(BT)

Term work: 50 Marks
Practical: 50 Marks
Design and Analysis of Algorithms (410241)
Machine Learning(410242)
Blockchain Technology(410243)

# Assignment No: 1

**Title of the Assignment:** Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyze their time and space complexity.

**Objective of the Assignment:** Students should be able to perform non-recursive and recursive programs to calculate Fibonacci numbers and analyze their time and space complexity.

---

**Prerequisite:**

1. Basic of Python or Java Programming
2. Concept of Recursive and Non-recursive functions
3. Execution flow of calculate Fibonacci numbers
4. Basic of Time and Space complexity

---

**Contents for Theory:**

1. **Introduction to Fibonacci numbers**
2. **Time and Space complexity**

**Introduction to Fibonacci numbers**

- The Fibonacci series, named after Italian mathematician Leonardo Pisano Bogollo, later known as Fibonacci, is a series (sum) formed by Fibonacci numbers denoted as Fn. The numbers in Fibonacci sequence are given as: 0, 1, 1, 2, 3, 5, 8, 13, 21, 38, . . .
- In a Fibonacci series, every term is the sum of the preceding two terms, starting from 0 and 1 as first and second terms. In some old references, the term '0' might be omitted.

**What is the Fibonacci Series?**

- The Fibonacci series is the sequence of numbers (also called Fibonacci numbers), where every number is the sum of the preceding two numbers, such that the first two terms are '0' and '1'.
- In some older versions of the series, the term '0' might be omitted. A Fibonacci series can thus be given as, 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, . . . It can be thus be observed that every term can be calculated by adding the two terms before it.
- Given the first term, F0 and second term, F1 as '0' and '1', the third term here can

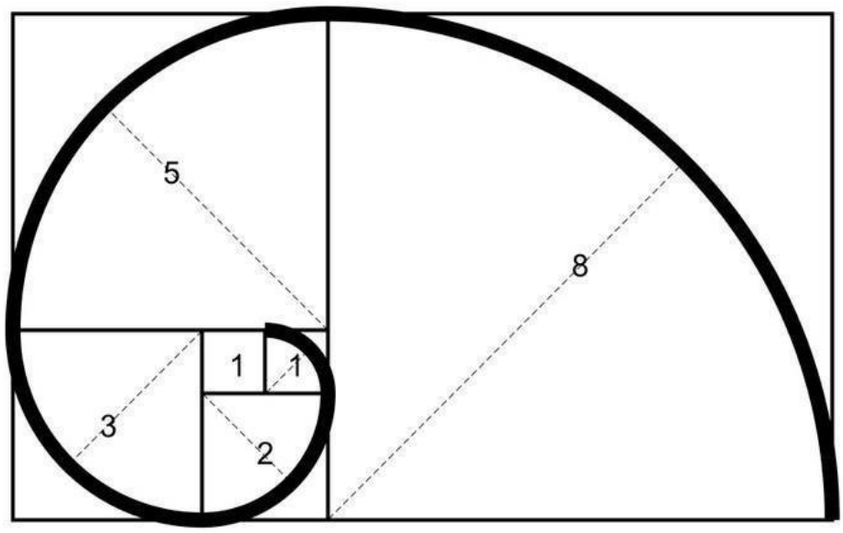
be given as, F2 = 0 + 1 = 1

Similarly,

F3 = 1 + 1 = 2

F4 = 2 + 1 = 3

Given a number n, print n-th Fibonacci Number.




## Fibonacci Sequence Formula

The Fibonacci sequence of numbers “Fn” is defined using the recursive relation with the seed values F0=0 and F1=1:

Fn = Fn-1+Fn-2

Here, the sequence is defined using two different parts, such as kick-off and recursive relation.

The kick-off part is F0=0 and F1=1.

The recursive relation part is Fn = Fn-1+Fn-2.

It is noted that the sequence starts with 0 rather than 1. So, F5 should be the 6th term of the sequence.

### Examples:

Input : n = 2

Output : 1

Input : n = 9

Output : 34

The list of Fibonacci numbers are calculated as follows:

| $F_n$ | Fibonacci Number |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |

| | |
|---|---|
| 5 | 5 |
| 6 | 8 |
| 7 | 13 |
| 8 | 21 |
| 9 | 34 |
| … and so on. | … and so on. |

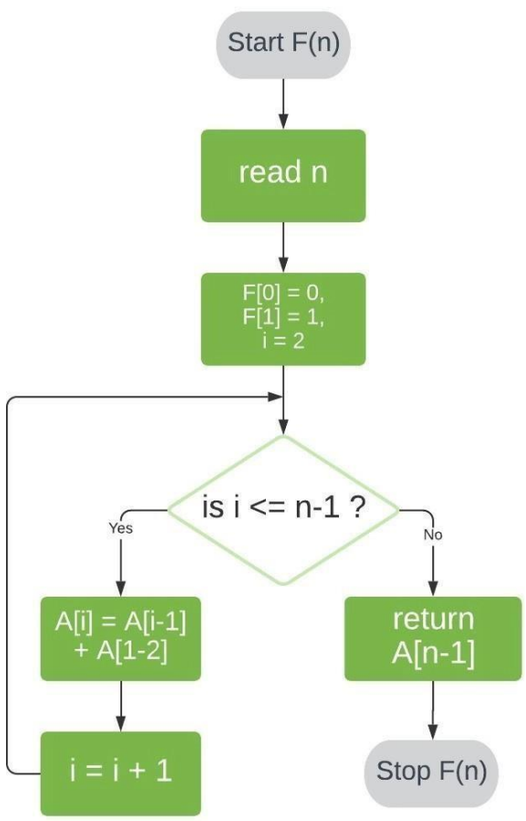
## Method 1 (Use Non-recursion)

A simple method that is a direct recursive implementation of mathematical recurrence relation is given above.

First, we'll store 0 and 1 in $F[0]$ and $F[1]$, respectively.

Next, we'll iterate through array positions 2 to $n-1$. At each position $i$, we store the sum of the two preceding array values in $F[i]$.

Finally, we return the value of $F[n-1]$, giving us the number at position $n$ in the sequence. Here's a visual representation of this process:



**Time and Space Complexity of Space Optimized Method**

- The time complexity of the Fibonacci series is **T(N) i.e, linear**. We have to find the sum of two terms and it is repeated n times depending on the value of n.
- The space complexity of the Fibonacci series using dynamic programming is **O(1)**.

**Time Complexity and Space Complexity of Dynamic Programming**

- The time complexity of the above code is **T(N) i.e, linear**. We have to find the sum of two terms and it is repeated n times depending on the value of n.
- The space complexity of the above code is **O(N)**.

## Method 2 (Use Recursion)

Let's start by defining F(n) as the function that returns the value of Fn.

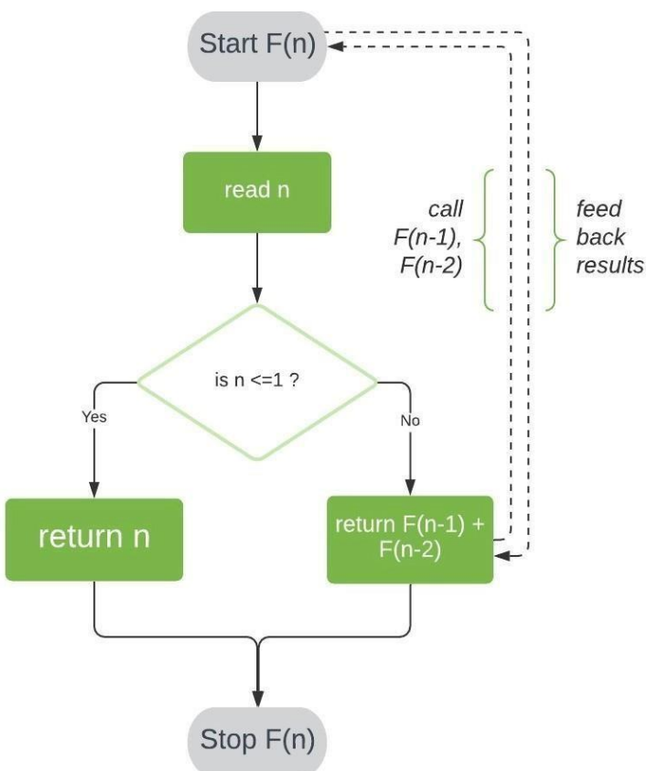
**To evaluate $F(n)$ for $n > 1$, we can reduce our problem into two smaller problems of the same kind: $F(n-1)$ and $F(n-2)$.** We can further reduce $F(n-1)$ and $F(n-2)$ to $F((n-1)-1)$ and $F((n-1)-2)$; and $F((n-2)-1)$ and $F((n-2)-2)$, respectively.

If we repeat this reduction, we'll eventually reach our known base cases and, thereby, obtain a solution to $F(n)$.

Employing this logic, our algorithm for $F(n)$ will have two steps:

1. Check if $n \leq 1$. If so, return $n$.
2. Check if $n > 1$. If so, call our function $F$ with inputs $n-1$ and $n-2$, and return the sum of the two results.

Here's a visual representation of this algorithm:



**Time and Space Complexity**

- The time complexity of the above code is **T(2^N) i.e, exponential**.

- The Space complexity of the above code is **O(N) for a recursive series.**

| Method | Time complexity | Space complexity |
|---|---|---|
| **Using recursion** | **T(n) = T(n-1) + T(n-2)** | **O(n)** |
| **Using DP** | **O(n)** | **O(1)** |
| **Space optimization of DP** | **O(n)** | **O(1)** |
| **Using the power of matrix method** | **O(n)** | **O(1)** |
| **Optimized matrix method** | **O(log n)** | **O(log n)** |
| **Recursive method in O(log n) time** | **O(log n)** | **O(n)** |
| **Using direct formula** | **O(log n)** | **O(1)** |
| **DP using memoization** | **O(n)** | **O(1)** |

**Applications of Fibonacci Series**

The Fibonacci series finds application in different fields in our day-to-day lives. The different patterns found in a varied number of fields from nature, to music, and to the human body follow the Fibonacci series. Some of the applications of the series are given as,

- It is used in the grouping of numbers and used to study different other special mathematical sequences.
- It finds application in Coding (computer algorithms, distributed systems, etc). For example, Fibonacci series are important in the computational run-time analysis of Euclid's algorithm, used for determining the GCF of two integers.
- It is applied in numerous fields of science like quantum mechanics, cryptography, etc.
- In finance market trading, Fibonacci retracement levels are widely used in technical analysis.

**Conclusion**- In this way we have explored Concept of Fibonacci series using recursive and non-recursive method and also learn time and space complexity.

# Assignment No: 2

**Title of the Assignment:** Write a program to implement Huffman Encoding using a greedy strategy.

**Objective of the Assignment:** Students should be able to understand and solve Huffman Encoding using greedy method

**Prerequisite:**

1. Basic of Python or Java Programming
2. Concept of Greedy method
3. Huffman Encoding concept

---

**Contents for Theory:**

1. **Greedy Method**
2. **Huffman Encoding**
3. **Example solved using huffman encoding**

---

**What is a Greedy Method?**

- A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't worry whether the current best result will bring the overall optimal result.
- The algorithm never reverses the earlier decision even if the choice is wrong. It works in a top-down approach.
- This algorithm may not produce the best result for all the problems. It's because it always goes for the local best choice to produce the global best result.

**Advantages of Greedy Approach**

- The algorithm is **easier to describe**.
- This algorithm can **perform better** than other algorithms (but, not in all cases).

**Drawback of Greedy Approach**

- As mentioned earlier, the greedy algorithm doesn't always produce the optimal solution. This is the major disadvantage of the algorithm
- For example, suppose we want to find the longest path in the graph below from root to leaf.

**Greedy Algorithm**

1. To begin with, the solution set (containing answers) is empty.
2. At each step, an item is added to the solution set until a solution is reached.

3. If the solution set is feasible, the current item is kept.
4. Else, the item is rejected and never considered again.

**Huffman Encoding**

- Huffman Coding is a technique of compressing data to reduce its size without losing any of the details. It was first developed by David Huffman.
- Huffman Coding is generally useful to compress the data in which there are frequently occurring characters.
- Huffman Coding is a famous Greedy Algorithm.
- It is used for the lossless compression of data.
- It uses variable length encoding.
- It assigns variable length code to all the characters.
- The code length of a character depends on how frequently it occurs in the given text.
- The character which occurs most frequently gets the smallest code.
- The character which occurs least frequently gets the largest code.
- It is also known as **Huffman Encoding**.

**Prefix Rule-**

- Huffman Coding implements a rule known as a prefix rule.
- This is to prevent the ambiguities while decoding.
- It ensures that the code assigned to any character is not a prefix of the code assigned to any other character

**Major Steps in Huffman Coding-**

There are two major steps in Huffman Coding-

1. Building a Huffman Tree from the input characters.
2. Assigning code to the characters by traversing the Huffman Tree.

**How does Huffman Coding work?**

Suppose the string below is to be sent over a network.


Initial string

- Each character occupies 8 bits. There are a total of 15 characters in the above string. Thus, a total of 8 * 15 = 120 bits are required to send this string.
- Using the Huffman Coding technique, we can compress the string to a smaller size.
- Huffman coding first creates a tree using the frequencies of the character and then generates code for each character.
- Once the data is encoded, it has to be decoded. Decoding is done using the same tree.

- Huffman Coding prevents any ambiguity in the decoding process using the concept of **prefix code** ie. a code associated with a character should not be present in the prefix of any other code. The tree created above helps in maintaining the property.
- Huffman coding is done with the help of the following steps.

1. Calculate the frequency of each character in the string.

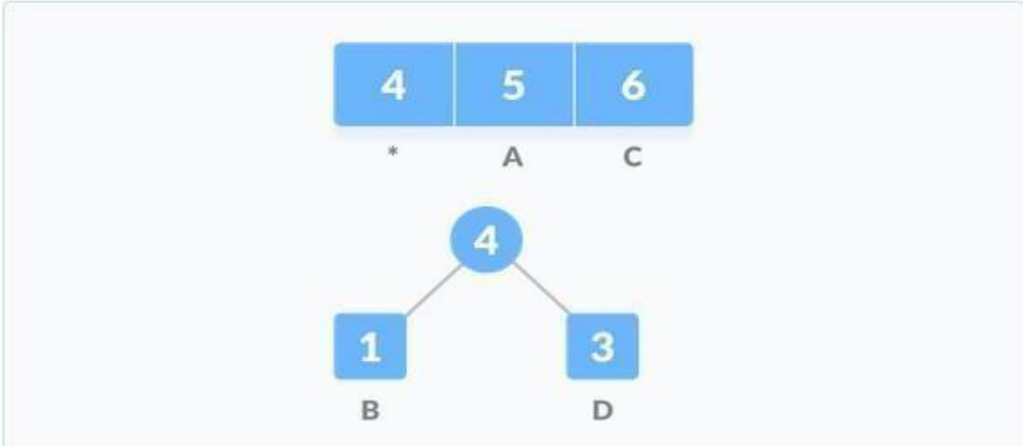

Frequency of string

2. Sort the characters in increasing order of the frequency. These are stored in a priority queue Q.


Characters sorted according to the frequency

3. Make each unique character as a leaf node.

4. Create an empty node z. Assign the minimum frequency to the left child of z and assign the second minimum frequency to the right child of z. Set the value of the z as the sum of the above two minimum frequencies.

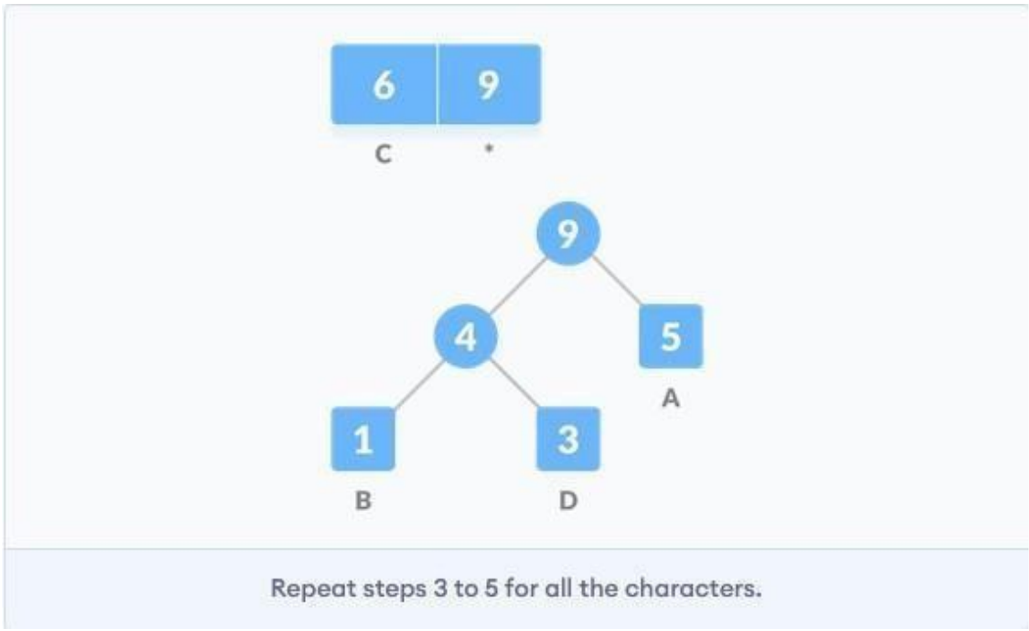



Getting the sum of the least numbers

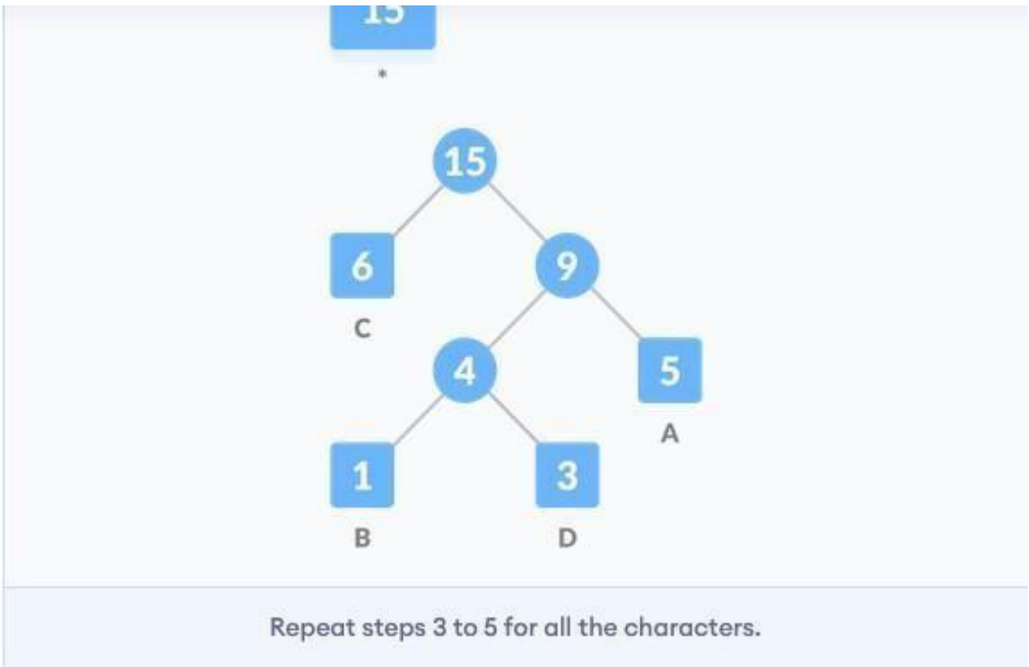
5. Remove these two minimum frequencies from Q and add the sum into the list of frequencies (* denote the internal nodes in the figure above).

6. Insert node z into the tree.

7. Repeat steps 3 to 5 for all the characters.

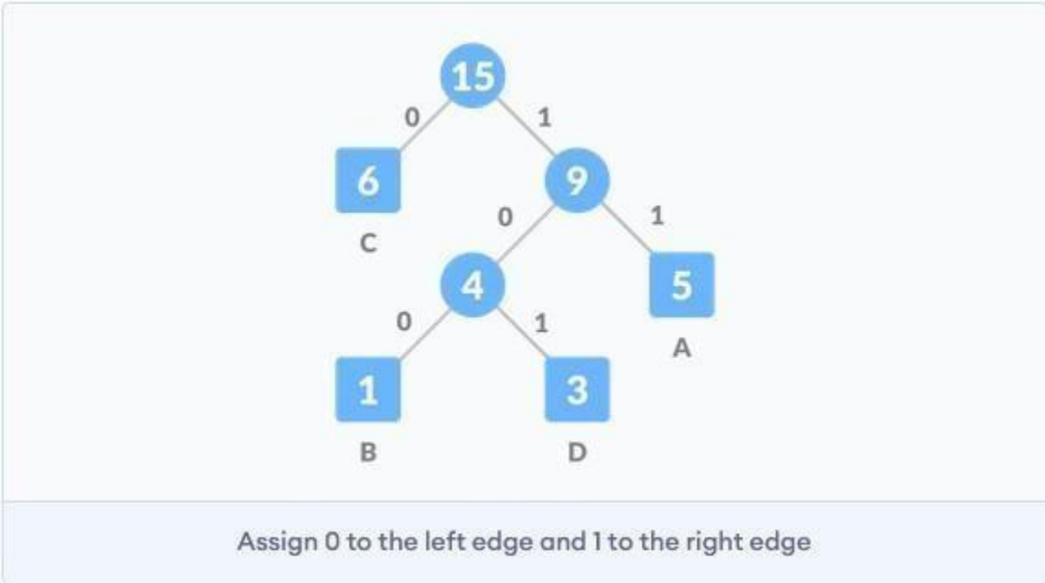



Repeat steps 3 to 5 for all the characters.




Repeat steps 3 to 5 for all the characters.

8. For each non-leaf node, assign 0 to the left edge and 1 to the right edge




Assign 0 to the left edge and 1 to the right edge

For sending the above string over a network, we have to send the tree as well as the above compressed-code. The total size is given by the table below.
Without encoding, the total size of the string was 120 bits. After encoding the size is reduced to 32 + 15 + 28 = 75.

| Character | Frequency | Code | Size |
|---|---|---|---|
| A | 5 | 11 | 5*2 = 10 |
| B | 1 | 100 | 1*3 = 3 |
| C | 6 | 0 | 6*1 = 6 |
| D | 3 | 101 | 3*3 = 9 |
| 4 * 8 = 32 bits | 15 bits | | 28 bits |

**Example:**

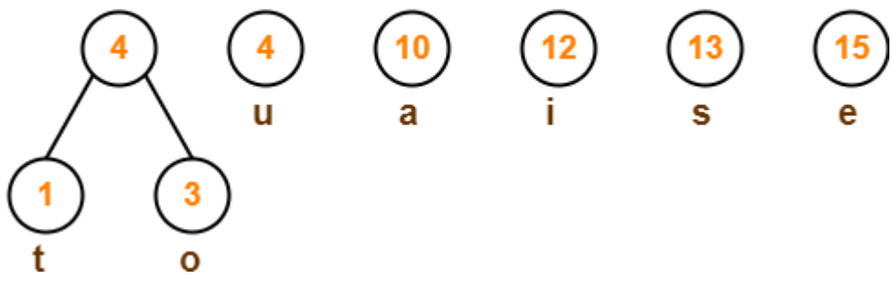
A file contains the following characters with the frequencies as shown. If Huffman Coding is used for data compression, determine-

1. Huffman Code for each character
2. Average code length
3. Length of Huffman encoded message (in bits)

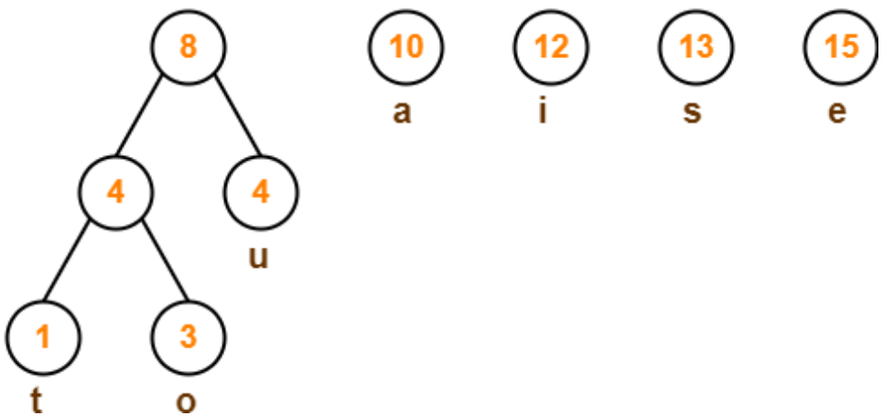
| Characters | Frequencies |
|---|---|
| a | 10 |
| e | 15 |
| i | 12 |
| o | 3 |
| u | 4 |
| s | 13 |
| t | 1 |

**Step-01:**

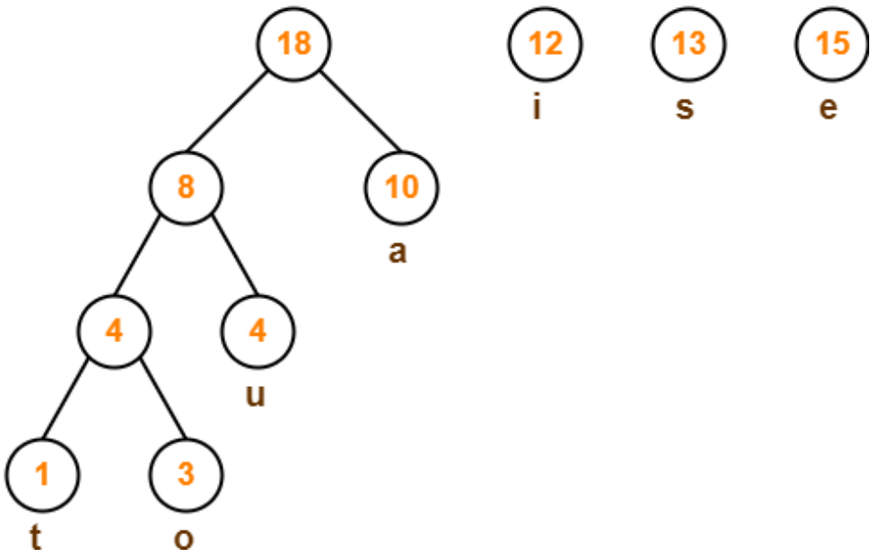

**Step-02:**



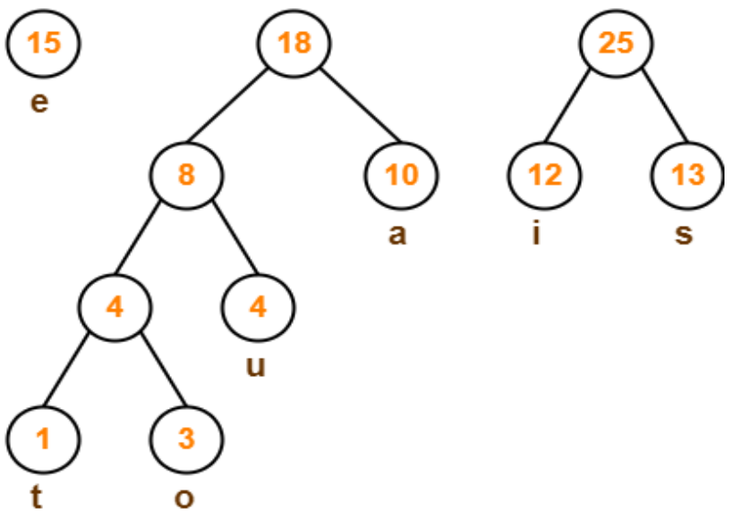

**Step-03:**

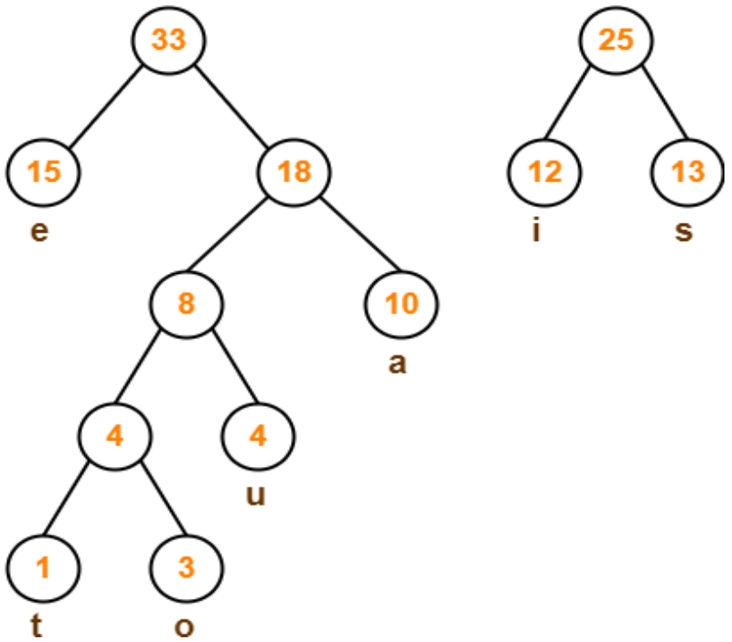


**Step-04:**

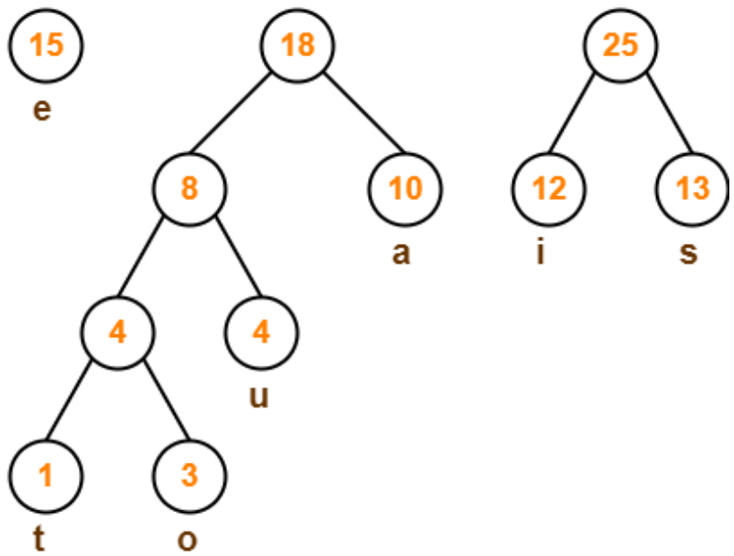


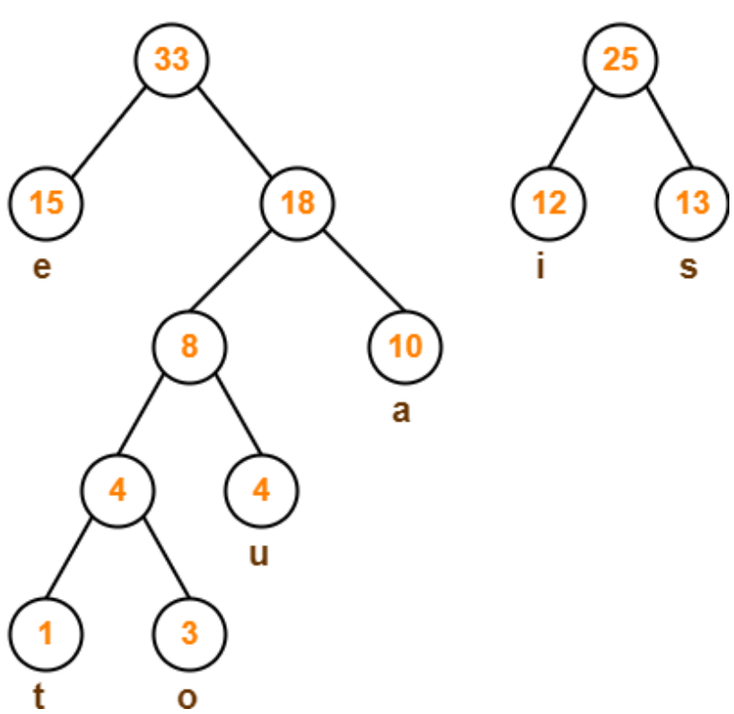

**Step-06:**



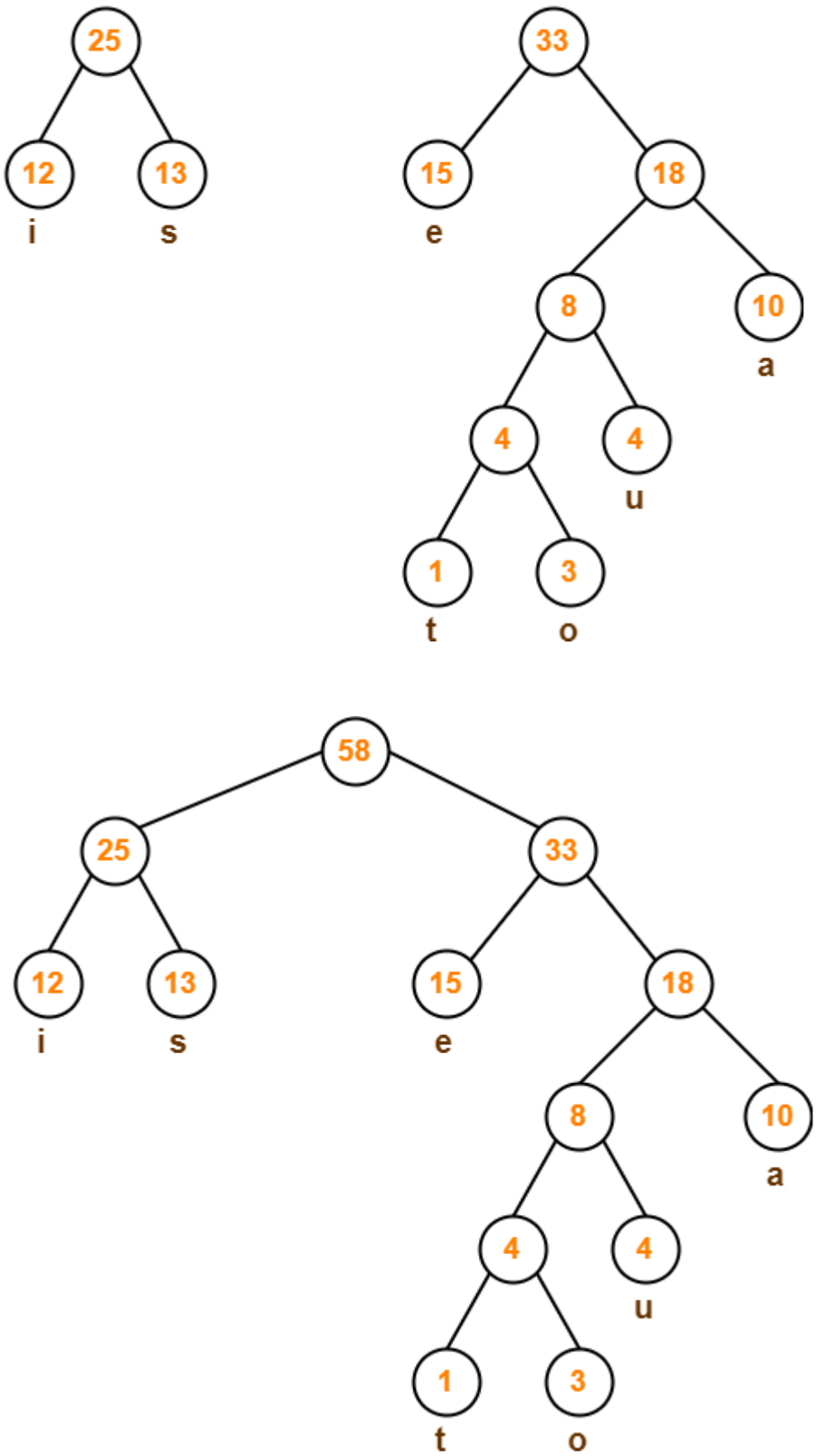



**Step-06:**

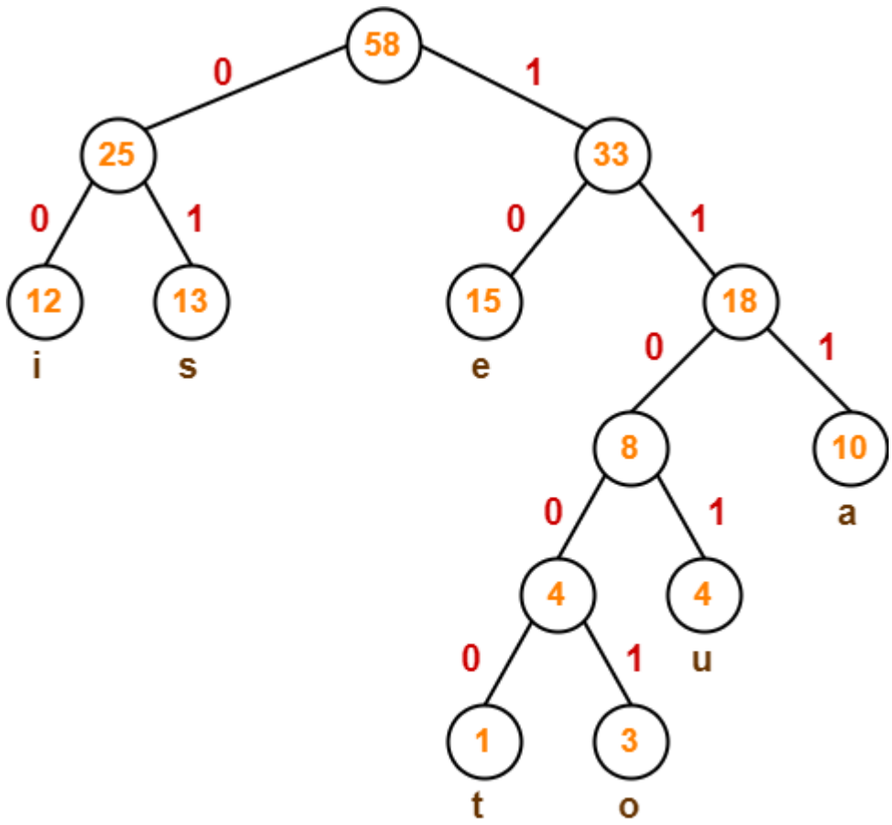

15
e
18
8
10
a
25
12
13
i
s
4
4
u
1
3
t
o


33
15
e
18
8
10
a
25
12
13
i
s
4
4
u
1
3
t
o

**Step-07:**




After assigning weight to all the edges, the modified Huffman Tree is-




**Huffman Tree**

To write Huffman Code for any character, traverse the Huffman Tree from root node to the leaf node of that character.

Following this rule, the Huffman Code for each character is-

a = 111

e = 10

i = 00

o = 11001

u = 1101

s = 01

t = 11000

**Time Complexity-**

The time complexity analysis of Huffman Coding is as follows-

- extractMin( ) is called 2 x (n-1) times if there are n nodes.
- As extractMin( ) calls minHeapify( ), it takes O(logn) time.

Thus, Overall time complexity of Huffman Coding becomes **O(nlogn)**.

**Conclusion**- In this way we have explored Concept of Huffman Encoding using greedy method.

# Assignment No: 3

**Title of the Assignment:** Write a program to solve a fractional Knapsack problem using a greedy method.

**Objective of the Assignment:** Students should be able to understand and solve fractional Knapsack problems using a greedy method.

**Prerequisite:**

1. Basic of Python or Java Programming
2. Concept of Greedy method
3. fractional Knapsack problem

---

**Contents for Theory:**

1. **Greedy Method**
2. **Fractional Knapsack problem**
3. **Example solved using fractional Knapsack problem**

---

**What is a Greedy Method?**

- A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't worry whether the current best result will bring the overall optimal result.
- The algorithm never reverses the earlier decision even if the choice is wrong. It works in a top-down approach.
- This algorithm may not produce the best result for all the problems. It's because it always goes for the local best choice to produce the global best result.

**Advantages of Greedy Approach**

- The algorithm is **easier to describe**.
- This algorithm can **perform better** than other algorithms (but, not in all cases).

**Drawback of Greedy Approach**

- As mentioned earlier, the greedy algorithm doesn't always produce the optimal solution. This is the major disadvantage of the algorithm
- For example, suppose we want to find the longest path in the graph below from root to leaf.

**Greedy Algorithm**

1. To begin with, the solution set (containing answers) is empty.
2. At each step, an item is added to the solution set until a solution is reached.
3. If the solution set is feasible, the current item is kept.

4. Else, the item is rejected and never considered again.

**Knapsack Problem**

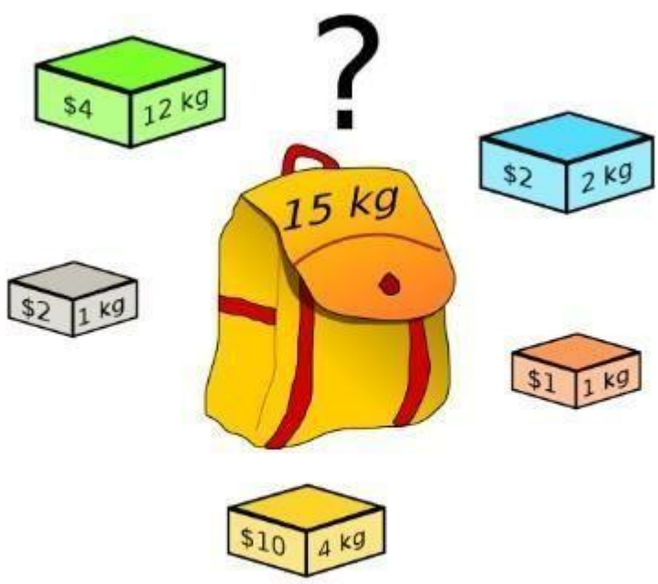
You are given the following-

- A knapsack (kind of shoulder bag) with limited weight capacity.
- Few items each having some weight and value.

The problem states-

Which items should be placed into the knapsack such that-

- The value or profit obtained by putting the items into the knapsack is maximum.
- And the weight limit of the knapsack does not exceed.




Knapsack Problem

## Knapsack Problem Variants

Knapsack problem has the following two variants-

1. Fractional Knapsack Problem
2. 0/1 Knapsack Problem

## Fractional Knapsack Problem-

In Fractional Knapsack Problem,

- As the name suggests, items are divisible here.
- We can even put the fraction of any item into the knapsack if taking the complete item is not
- possible.

It is solved using the Greedy Method.

# Fractional Knapsack Problem Using Greedy Method-

Fractional knapsack problem is solved using greedy method in the following steps-

**Step-01:**

For each item, compute its value / weight ratio.

**Step-02:**

Arrange all the items in decreasing order of their value / weight ratio.

**Step-03:**

Start putting the items into the knapsack beginning from the item with the highest ratio.

Put as many items as you can into the knapsack.

# Problem-

For the given set of items and knapsack capacity = 60 kg, find the optimal solution for the fractional knapsack problem making use of greedy approach.

| Item | Weight | Value |
|---|---|---|
| 1 | 5 | 30 |
| 2 | 10 | 40 |
| 3 | 15 | 45 |
| 4 | 22 | 77 |
| 5 | 25 | 90 |

n = 5

w = 60 kg

(w1, w2, w3, w4, w5) = (5, 10, 15, 22, 25)

(b1, b2, b3, b4, b5) = (30, 40, 45, 77, 90)

## Solution-

### Step-01:

Compute the value / weight ratio for each item-

| Items | Weight | Value | Ratio |
|---|---|---|---|
| 1 | 5 | 30 | 6 |
| 2 | 10 | 40 | 4 |
| 3 | 15 | 45 | 3 |
| 4 | 22 | 77 | 3.5 |
| 5 | 25 | 90 | 3.6 |

**Step-02:**

Sort all the items in decreasing order of their value / weight ratio-

**I1 I2 I5 I4 I3**

(6) (4) (3.6) (3.5) (3)

**Step-03:**

Start filling the knapsack by putting the items into it one by one.

| Knapsack Weight | Items in Knapsack | Cost |
|---|---|---|
| 60 | Ø | 0 |
| 55 | I1 | 30 |
| 45 | I1, I2 | 70 |
| 20 | I1, I2, I5 | 160 |

Now,

- Knapsack weight left to be filled is 20 kg but item-4 has a weight of 22 kg.
- Since in fractional knapsack problem, even the fraction of any item can be taken.
- So, knapsack will contain the following items-

< I1 , I2 , I5 , (20/22) I4 >

**Total cost of the knapsack**

= 160 + (20/22) x 77

= 160 + 70

= 230 units

## Time Complexity-

- The main time taking step is the sorting of all items in decreasing order of their value / weight ratio.
- If the items are already arranged in the required order, then while loop takes O(n) time.
- The average time complexity of Quick Sort is O(nlogn).
- Therefore, total time taken including the sort is O(nlogn).

**Conclusion**-In this way we have explored Concept of Fractional Knapsack using greedy method

# Assignment No: 4

**Title of the Assignment:** Write a program to solve a 0-1 Knapsack problem using dynamic programming or branch and bound strategy.

**Objective of the Assignment:** Students should be able to understand and solve 0-1 Knapsack problem using dynamic programming

**Prerequisite:**

1. Basic of Python or Java Programming
2. Concept of Dynamic Programming
3. 0/1 Knapsack problem

---

**Contents for Theory:**

1. **Greedy Method**
2. **0/1 Knapsack problem**
3. **Example solved using 0/1 Knapsack problem**

---

**What is Dynamic Programming?**

- Dynamic Programming is also used in optimization problems. Like divide-and-conquer method, Dynamic Programming solves problems by combining the solutions of subproblems.
- Dynamic Programming algorithm solves each sub-problem just once and then saves its answer in a table, thereby avoiding the work of re-computing the answer every time.
- Two main properties of a problem suggest that the given problem can be solved using Dynamic Programming. These properties are **overlapping sub-problems and optimal substructure**.
- Dynamic Programming also combines solutions to sub-problems. It is mainly used where the solution of one sub-problem is needed repeatedly. The computed solutions are stored in a table, so that these don‟t have to be re-computed. Hence, this technique is needed where overlapping sub-problem exists.
- For example, Binary Search does not have overlapping sub-problem. Whereas recursive program of Fibonacci numbers have many overlapping sub-problems.

**Steps of Dynamic Programming Approach**

Dynamic Programming algorithm is designed using the following four steps −

- Characterize the structure of an optimal solution.
- Recursively define the value of an optimal solution.
- Compute the value of an optimal solution, typically in a bottom-up fashion.
- Construct an optimal solution from the computed information.

**Applications of Dynamic Programming Approach**

- Matrix Chain Multiplication
- Longest Common Subsequence
- Travelling Salesman Problem

**Knapsack Problem**

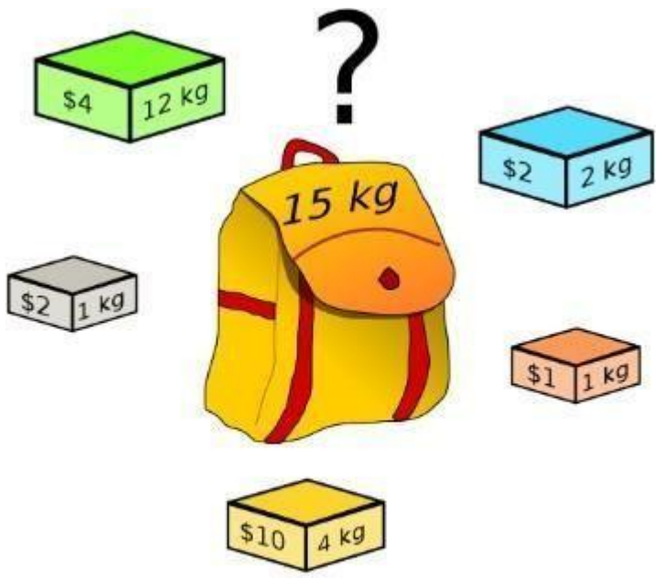
You are given the following-

- A knapsack (kind of shoulder bag) with limited weight capacity.
- Few items each having some weight and value.

The problem states-

Which items should be placed into the knapsack such that-

- The value or profit obtained by putting the items into the knapsack is maximum.
- And the weight limit of the knapsack does not exceed.




Knapsack Problem

## Knapsack Problem Variants

Knapsack problem has the following two variants-

1. Fractional Knapsack Problem
2. 0/1 Knapsack Problem

## 0/1 Knapsack Problem-

In 0/1 Knapsack Problem,

- As the name suggests, items are indivisible here.
- We can not take a fraction of any item.
- We have to either take an item completely or leave it completely.
- It is solved using a dynamic programming approach.

# 0/1 Knapsack Problem Using Greedy Method-

Consider-

- Knapsack weight capacity = w
- Number of items each having some weight and value = n

**0/1 knapsack problem is solved using dynamic programming in the following steps-**

## Step-01:

- Draw a table say „T" with (n+1) number of rows and (w+1) number of columns.
- Fill all the boxes of $0^{th}$ row and $0^{th}$ column with zeroes as shown-

| | 0 | 1 | 2 | 3 | ..... | W |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | ..... | 0 |
| 1 | 0 | | | | | |
| 2 | 0 | | | | | |
| | ..... | | | | | |
| n | 0 | | | | | |

T-Table

## Step-02:

Start filling the table row wise top to bottom from left to right.

Use the following formula-

$$T(i, j) = \max \{ T(i-1, j), value_i + T(i-1, j - weight_i) \}$$

Here, T(i , j) = maximum value of the selected items if we can take items 1 to i and have weight restrictions of j.

- This step leads to completely filling the table.
- Then, value of the last box represents the maximum possible value that can be put into the knapsack.

## Step-03:

- To identify the items that must be put into the knapsack to obtain that maximum profit,
- Consider the last column of the table.
- Start scanning the entries from bottom to top.
- On encountering an entry whose value is not same as the value stored in the entry immediately above it, mark the row label of that entry.
- After all the entries are scanned, the marked labels represent the items that must be put into the knapsack

Problem-.

For the given set of items and knapsack capacity = 5 kg, find the optimal solution for the 0/1 knapsack problem making use of a dynamic programming approach.

| Item | Weight | Value |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 4 |
| 3 | 4 | 5 |
| 4 | 5 | 6 |

n = 4

w = 5 kg

(w1, w2, w3, w4) = (2, 3, 4, 5)

(b1, b2, b3, b4) = (3, 4, 5, 6)

# Solution-

## Given

- Knapsack capacity (w) = 5 kg
- Number of items (n) = 4

## Step-01:

- Draw a table say „T" with (n+1) = 4 + 1 = 5 number of rows and (w+1) = 5 + 1 = 6 number of columns.
- Fill all the boxes of $0^{th}$ row and $0^{th}$ column with 0.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | | | | | |
| 2 | 0 | | | | | |
| 3 | 0 | | | | | |
| 4 | 0 | | | | | |

T-Table

## Step-02:

Start filling the table row wise top to bottom from left to right using the formula-

$$T(i, j) = \max \{ T(i-1, j), value_i + T(i-1, j - weight_i) \}$$

**Finding T(1,1)-**

We have,

- i = 1
- j = 1
- $(value)_i = (value)_1 = 3$
- $(weight)_i = (weight)_1 = 2$

Substituting the values, we get-

T(1,1) = max { T(1-1 , 1) , 3 + T(1-1 , 1-2) }

T(1,1) = max { T(0,1) , 3 + T(0,-1) }

T(1,1) = T(0,1) { Ignore T(0,-1) }

T(1,1) = 0

**Finding T(1,2)-**

We have,

- i = 1
- j = 2
- $(value)_i = (value)_1 = 3$
- $(weight)_i = (weight)_1 = 2$

Substituting the values, we get-

T(1,2) = max { T(1-1 , 2) , 3 + T(1-1 , 2-2) }

T(1,2) = max { T(0,2) , 3 + T(0,0) }

T(1,2) = max {0 , 3+0}

T(1,2) = 3

**Finding T(1,3)-**

We have,

- i = 1
- j = 3
- $(value)_i = (value)_1 = 3$
- $(weight)_i = (weight)_1 = 2$

Substituting the values, we get-

T(1,3) = max { T(1-1 , 3) , 3 + T(1-1 , 3-2) }

T(1,3) = max { T(0,3) , 3 + T(0,1) }

T(1,3) = max {0 , 3+0}

T(1,3) = 3

**Finding T(1,4)-**

We have,

- i = 1
- j = 4
- $(value)_i = (value)_1 = 3$
- $(weight)_i = (weight)_1 = 2$

Substituting the values, we get-

T(1,4) = max { T(1-1 , 4) , 3 + T(1-1 , 4-2) }

T(1,4) = max { T(0,4) , 3 + T(0,2) }

T(1,4) = max {0 , 3+0}

T(1,4) = 3

**Finding T(1,5)-**

We have,

- i = 1
- j = 5
- $(value)_i = (value)_1 = 3$
- $(weight)_i = (weight)_1 = 2$

Substituting the values, we get-

T(1,5) = max { T(1-1 , 5) , 3 + T(1-1 , 5-2) }

T(1,5) = max { T(0,5) , 3 + T(0,3) }

T(1,5) = max {0 , 3+0}

T(1,5) = 3

**Finding T(2,1)-**

We have,

- i = 2
- j = 1
- $(value)_i = (value)_2 = 4$
- (weight)i = (weight)2 = 3

Substituting the values, we get-

T(2,1) = max { T(2-1 , 1) , 4 + T(2-1 , 1-3) }

T(2,1) = max { T(1,1) , 4 + T(1,-2) }

T(2,1) = T(1,1) { Ignore T(1,-2) }

T(2,1) = 0

**Finding T(2,2)-**

We have,

- i = 2
- j = 2
- $(value)_i = (value)_2 = 4$
- $(weight)_i = (weight)_2 = 3$

Substituting the values, we get-

T(2,2) = max { T(2-1 , 2) , 4 + T(2-1 , 2-3) }

T(2,2) = max { T(1,2) , 4 + T(1,-1) }

T(2,2) = T(1,2) { Ignore T(1,-1) }

T(2,2) = 3

**Finding T(2,3)-**

We have,

- i = 2
- j = 3
- $(value)_i = (value)_2 = 4$
- $(weight)_i = (weight)_2 = 3$

Substituting the values, we get-

T(2,3) = max { T(2-1 , 3) , 4 + T(2-1 , 3-3) }

T(2,3) = max { T(1,3) , 4 + T(1,0) }

T(2,3) = max { 3 , 4+0 }

T(2,3) = 4

Similarly, compute all the entries.

After all the entries are computed and filled in the table, we get the following table-

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ✓ 1 | 0 | 0 | 3 | 3 | 3 | 3 |
| ✓ 2 | 0 | 0 | 3 | 4 | 4 | 7 |
| 3 | 0 | 0 | 3 | 4 | 5 | 7 |
| 4 | 0 | 0 | 3 | 4 | 5 | 7 |

T-Table

- The last entry represents the maximum possible value that can be put into the knapsack.
- So, maximum possible value that can be put into the knapsack = 7.

## Identifying Items To Be Put Into Knapsack

Following Step-04,

- We mark the rows labelled "1" and "2".
- Thus, items that must be put into the knapsack to obtain the maximum value 7 are-

**Item-1 and Item-2**

## Time Complexity-

- Each entry of the table requires constant time θ(1) for its computation.
- It takes θ(nw) time to fill (n+1)(w+1) table entries.
- It takes θ(n) time for tracing the solution since tracing process traces the n rows.
- Thus, overall θ(nw) time is taken to solve 0/1 knapsack problem using dynamic programming

**Conclusion-**In this way we have explored Concept of 0/1 Knapsack using Dynamic approch

In [1]:
```
nterms = int(input("How many terms? "))

n1, n2 = 0, 1
count = 0

if nterms <= 0:
   print("Please enter a positive integer")
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
else:
   print("Fibonacci sequence:")
   while count < nterms:
       print(n1)
       nth = n1 + n2
       # update values
       n1 = n2
       n2 = nth
       count += 1
```

```
Fibonacci sequence:
0
1
1
2
3
5
8
```

In [ ]:

In [ ]:

In [3]:

```
class Nodes:

    def __init__(self, probability, symbol, left = None, right = None):

        self.probability = probability

        self.symbol = symbol

        self.left = left

        self.right = right

        self.code = ''

def CalculateProbability(the_data):

    the_symbols = dict()

    for item in the_data:

        if the_symbols.get(item) == None:

            the_symbols[item] = 1
        else:
            the_symbols[item] += 1

    return the_symbols

the_codes = dict()

def CalculateCodes(node, value = ''):

    newValue = value + str(node.code)
    if(node.left):

        CalculateCodes(node.left, newValue)

    if(node.right):

        CalculateCodes(node.right, newValue)

    if(not node.left and not node.right):

        the_codes[node.symbol] = newValue

    return the_codes

def OutputEncoded(the_data, coding):

    encodingOutput = []

    for element in the_data:

        encodingOutput.append(coding[element])
```

```
    the_string = ''.join([str(item) for item in encodingOutput])
    return the_string
def TotalGain(the_data, coding):
    beforeCompression = len(the_data) * 8
    afterCompression = 0
    the_symbols = coding.keys()
    for symbol in the_symbols:
        the_count = the_data.count(symbol)
        afterCompression += the_count * len(coding[symbol])
    print("Space usage before compression (in bits):", beforeCompression)
    print("Space usage after compression (in bits):",  afterCompression)
def HuffmanEncoding(the_data):
    symbolWithProbs = CalculateProbability(the_data)
    the_symbols = symbolWithProbs.keys()
    the_probabilities = symbolWithProbs.values()
    print("symbols: ", the_symbols)
    print("probabilities: ", the_probabilities)
    the_nodes = []
    for symbol in the_symbols:
        the_nodes.append(Nodes(symbolWithProbs.get(symbol), symbol))
    while len(the_nodes) > 1:
        the_nodes = sorted(the_nodes, key = lambda x: x.probability)
        right = the_nodes[0]
        left = the_nodes[1]
        left.code = 0
        right.code = 1
        newNode = Nodes(left.probability + right.probability, left.symbol + right.s
        the_nodes.remove(left)
```

```
        the_nodes.remove(right)
        the_nodes.append(newNode)
    huffmanEncoding = CalculateCodes(the_nodes[0])
    print("symbols with codes", huffmanEncoding)
    TotalGain(the_data, huffmanEncoding)
    encodedOutput = OutputEncoded(the_data,huffmanEncoding)
    return encodedOutput, the_nodes[0]
def HuffmanDecoding(encodedData, huffmanTree):
    treeHead = huffmanTree
    decodedOutput = []
    for x in encodedData:
        if x == '1':
            huffmanTree = huffmanTree.right
        elif x == '0':
            huffmanTree = huffmanTree.left
        try:
            if huffmanTree.left.symbol == None and huffmanTree.right.symbol == None
                pass
        except AttributeError:
            decodedOutput.append(huffmanTree.symbol)
            huffmanTree = treeHead

    string = ''.join([str(item) for item in decodedOutput])
    return string

the_data = "AAAAAAABBCCCCCCDDDEEEEEEEEE"
print(the_data)
encoding, the_tree = HuffmanEncoding(the_data)
print("Encoded output", encoding)
```

```
print("Decoded Output", HuffmanDecoding(encoding, the_tree))
```

```
AAAAAAABBCCCCCCDDDEEEEEEEEE
symbols:  dict_keys(['A', 'B', 'C', 'D', 'E'])
probabilities:  dict_values([7, 2, 6, 3, 9])
symbols with codes {'E': '00', 'A': '01', 'C': '10', 'D': '110', 'B': '111'}
Space usage before compression (in bits): 216
Space usage after compression (in bits): 59
Encoded output 01010101010101111111101010101011011011000000000000000000000
Decoded Output AAAAAAABBCCCCCCDDDEEEEEEEEE
```

In [ ]:

In [2]:
```python
class Item:
    def __init__(self, profit, weight):
        self.profit = profit
        self.weight = weight

def fractionalKnapsack(W, arr):
    arr.sort(key=lambda x: (x.profit/x.weight), reverse=True)

    finalvalue = 0.0

    for item in arr:
        if item.weight <= W:
            W -= item.weight
            finalvalue += item.profit
        else:
            finalvalue += item.profit * W / item.weight
            break
    return finalvalue
if __name__ == "__main__":
    W = 50
    arr = [Item(60, 10), Item(100, 20), Item(120, 30)]
    max_val = fractionalKnapsack(W, arr)
    print(max_val)
```

```
240.0
```

In [ ]:

In [5]:
```python
def knapsack(wt, val, W, n):
    if n == 0 or W == 0:
        return 0

    if t[n][W] != -1:
        return t[n][W]

    if wt[n - 1] <= W:
        t[n][W] = max(
            val[n - 1] + knapsack(wt, val, W - wt[n - 1], n - 1),
            knapsack(wt, val, W, n - 1)
        )
    else:
        t[n][W] = knapsack(wt, val, W, n - 1)

    return t[n][W]

if __name__ == '__main__':
    profit = [60, 100, 120]
    weight = [10, 20, 30]
    W = 50
    n = len(profit)

    t = [[-1 for _ in range(W + 1)] for _ in range(n + 1)]

    print("Maximum Profit:", knapsack(weight, profit, W, n))
```

```
Maximum Profit: 220
```

In [ ]:

# DAA MINI PROJECT

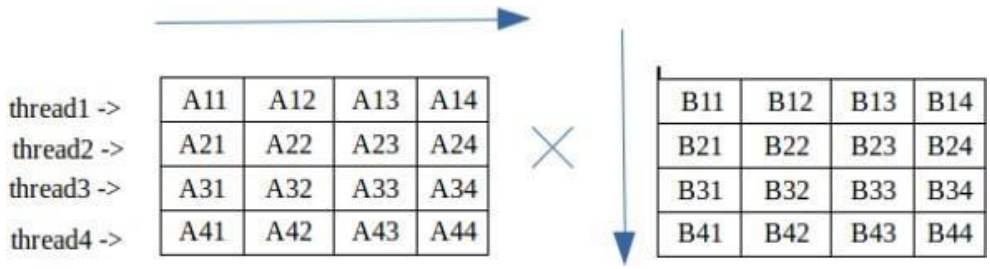
Multiplication of matrix does take time surely. Time complexity of matrix multiplication is O(n^3) using normal matrix multiplication. And Strassen algorithm improves it and its time complexity is O(n^(2.8074)).

But, Is there any way to improve the performance of matrix multiplication using the normal method.

Multi-threading can be done to improve it. In multi-threading, instead of utilizing a single core of your processor, we utilizes all or more core to solve the problem.
We create different threads, each thread evaluating some part of matrix multiplication.

Depending upon the number of cores your processor has, you can create the number of threads required. Although you can create as many threads as you need, a better way is to create each thread for one core.

In second approach,we create a separate thread for each element in resultant matrix. Using pthread_exit() we return computed value from each thread which is collected by pthread_join(). This approach does not make use of any global variables.




**Code :-**

```
// CPP Program to multiply two matrix using pthreads
#include <bits/stdc++.h>
using namespace std;

// maximum size of matrix
#define MAX 4

// maximum number of threads
#define MAX_THREAD 4

int  matA[MAX][MAX];
int  matB[MAX][MAX];
int  matC[MAX][MAX];
int step_i = 0;

void* multi(void* arg)
```

```
{
    int i = step_i++; //i denotes row number of resultant matC

    for (int j = 0; j < MAX; j++)
    for (int k = 0; k < MAX; k++)
            matC[i][j] += matA[i][k] * matB[k][j];
}

// Driver Code
int main()


{
    // Generating random values in matA and matB
    for (int i = 0; i < MAX; i++) {
            for (int j = 0; j < MAX; j++) {
                    matA[i][j] = rand() % 10;
                    matB[i][j] = rand() % 10;
            }

    }

    // Displaying matA
    cout << endl
            << "Matrix A" << endl;
    for (int i = 0; i < MAX; i++) {
            for (int j = 0; j < MAX; j++)
                    cout << matA[i][j] << " ";
            cout << endl;
    }

    // Displaying matB
    cout << endl
            << "Matrix B" << endl;
    for (int i = 0; i < MAX; i++) {
            for (int j = 0; j < MAX; j++)
                    cout << matB[i][j] << " ";
            cout << endl;
    }

    // declaring four threads
    pthread_t threads[MAX_THREAD];

    // Creating four threads, each evaluating its own part
    for (int i = 0; i < MAX_THREAD; i++) {
            int* p;
            pthread_create(&threads[i], NULL, multi, (void*)(p));
    }

    // joining and waiting for all threads to complete
    for (int i = 0; i < MAX_THREAD; i++)
            pthread_join(threads[i], NULL);

    // Displaying the result matrix
    cout << endl
            << "Multiplication of A and B" << endl;
    for (int i = 0; i < MAX; i++) {
            for (int j = 0; j < MAX; j++)
                    cout << matC[i][j] << " ";
            cout << endl;
    }
    return 0;
}
```

# Group B
# Assignment No : 2

**Title of the Assignment:**Classify the email using the binary classification method. Email Spam detection has two states:
a) Normal State - Not Spam,
b) Abnormal State - Spam.
Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance.

**Dataset Description:**The csv file contains 5172 rows, each row for each email. There are 3002 columns. The first column indicates Email name. The name has been set with numbers and not recipients' name to protect privacy. The last column has the labels for prediction : 1for spam, 0 for not spam. The remaining 3000 columns are the 3000 most common words inall the emails, after excluding the non-alphabetical characters/words. For each row, thecount of each word(column) in that email(row) is stored in the respective cells. Thus,information regarding all 5172 emails are stored in a compact dataframe rather than asseparate text files.

**Link:**https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv

**Objective of the Assignment:**

Students should be able to classify email using the binary Classification and implement email spam detection technique by using K-Nearest Neighbors and Support Vector Machine algorithm.

**Prerequisite:**

1. Basic knowledge of Python
2. Concept of K-Nearest Neighbors and Support Vector Machine for classification.

**Contents of the Theory:**

1. Data Preprocessing
2. Binary Classification
3. K-Nearest Neighbours
4. Support Vector Machine
5. Train, Test and Split Procedure

**Data Preprocessing:**

Data preprocessing is a process of preparing the raw data and making it suitable for amachine learning model. It is the rst and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean itand put in a formatted way. So for this, we use data preprocessing task.

Why do we need Data Preprocessing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and e ciency of a machine learning model.

It involves below steps:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

Code :- https://www.kaggle.com/code/mfaisalqureshi/email-spam-detection-98-accuracy/notebook

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report
```

In [2]:
```
# Load dataset
df = pd.read_csv('emails.csv')
df.head()
```

Out[2]:

| | Email No. | the | to | ect | and | for | of | a | you | hou | ... | connevey | jay | valued | lay | infrastructure |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Email 1 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| **1** | Email 2 | 8 | 13 | 24 | 6 | 6 | 2 | 102 | 1 | 27 | ... | 0 | 0 | 0 | 0 | 0 |
| **2** | Email 3 | 0 | 0 | 1 | 0 | 0 | 0 | 8 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| **3** | Email 4 | 0 | 5 | 22 | 0 | 5 | 1 | 51 | 2 | 10 | ... | 0 | 0 | 0 | 0 | 0 |
| **4** | Email 5 | 7 | 6 | 17 | 1 | 5 | 2 | 57 | 0 | 9 | ... | 0 | 0 | 0 | 0 | 0 |

5 rows × 3002 columns

In [3]:
```
df.shape
```

Out[3]:
```
(5172, 3002)
```

In [4]:
```
df.info
```

Out[4]:
```
<bound method DataFrame.info of        Email No.  the  to  ect  and  for  of    a  you  hou
...  connevey  \
0         Email 1    0   0    1    0    0   0    2    0    0  ...         0
1         Email 2    8  13   24    6    6   2  102    1   27  ...         0
2         Email 3    0   0    1    0    0   0    8    0    0  ...         0
3         Email 4    0   5   22    0    5   1   51    2   10  ...         0
4         Email 5    7   6   17    1    5   2   57    0    9  ...         0
...           ...  ...  ..  ...  ...  ...  ..  ...  ...  ...  ...       ...
5167  Email 5168    2   2    2    3    0   0   32    0    0  ...         0
5168  Email 5169   35  27   11    2    6   5  151    4    3  ...         0
5169  Email 5170    0   0    1    1    0   0   11    0    0  ...         0
5170  Email 5171    2   7    1    0    2   1   28    2    0  ...         0
5171  Email 5172   22  24    5    1    6   5  148    8    2  ...         0

      jay  valued  lay  infrastructure  military  allowing  ff  dry  \
0       0       0    0               0         0         0   0    0
1       0       0    0               0         0         0   1    0
2       0       0    0               0         0         0   0    0
3       0       0    0               0         0         0   0    0
4       0       0    0               0         0         0   1    0
...   ...     ...  ...             ...       ...       ...  ..  ...
5167    0       0    0               0         0         0   0    0
5168    0       0    0               0         0         0   1    0
5169    0       0    0               0         0         0   0    0
5170    0       0    0               0         0         0   1    0
5171    0       0    0               0         0         0   0    0

      Prediction
0              0
1              0
2              0
3              0
4              0
...          ...
5167           0
5168           0
5169           1
5170           1
5171           0

[5172 rows x 3002 columns]>
```

In [5]:
```
df.describe
```

Out[5]:
```
<bound method NDFrame.describe of        Email No.  the  to  ect  and  for  of    a  you  hou
...  connevey  \
0         Email 1    0   0    1    0    0   0    2    0    0  ...         0
1         Email 2    8  13   24    6    6   2  102    1   27  ...           0
2         Email 3    0   0    1    0    0   0    8    0    0  ...         0
3         Email 4    0   5   22    0    5   1   51    2   10  ...         0
4         Email 5    7   6   17    1    5   2   57    0    9  ...         0
...           ...  ...  ..  ...  ...  ...  ..  ...  ...  ...  ...       ...
5167  Email 5168    2   2    2    3    0   0   32    0    0  ...         0
5168  Email 5169   35  27   11    2    6   5  151    4    3  ...         0
5169  Email 5170    0   0    1    1    0   0   11    0    0  ...         0
5170  Email 5171    2   7    1    0    2   1   28    2    0  ...           0
5171  Email 5172   22  24    5    1    6   5  148    8    2  ...         0

      jay  valued  lay  infrastructure  military  allowing  ff  dry  \
0       0       0    0               0         0         0   0    0
1       0       0    0               0         0         0   1    0
2       0       0    0               0         0         0   0    0
3       0       0    0               0         0         0   0    0
4       0       0    0               0         0         0   1    0
...   ...     ...  ...             ...       ...       ...  ..  ...
5167    0       0    0               0         0         0   0    0
5168    0       0    0               0         0         0   1    0
5169    0       0    0               0         0         0   0    0
5170    0       0    0               0         0         0   1    0
5171    0       0    0               0         0         0   0    0

      Prediction
0              0
1              0
2              0
3              0
4              0
...          ...
5167           0
5168           0
5169           1
5170           1
5171           0

[5172 rows x 3002 columns]>
```

In [6]:
```
# Data preprocessing
df.drop(columns=['Email No.'], inplace=True)
df.drop_duplicates(inplace=True)
```

In [7]:
```
# Features and target
X = df.drop(columns='Prediction', axis=1)
y = df['Prediction']
```

In [8]:
```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=36)
```

In [9]:
```
# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [10]:
```
print("\n" + "="*50)
print("EMAIL SPAM CLASSIFICATION")
print("Normal State (0) = Ham | Abnormal State (1) = Spam")
print("="*50)
```

```
==================================================
EMAIL SPAM CLASSIFICATION
Normal State (0) = Ham | Abnormal State (1) = Spam
==================================================
```

In [11]:
```
# 1. K-Nearest Neighbors Classification
print("\n1. K-NEAREST NEIGHBORS (KNN)")
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train_scaled, y_train)
```

```
1. K-NEAREST NEIGHBORS (KNN)
```

Out[11]:

KNeighborsClassifier

▶ Parameters

In [12]:
```
knn_pred = knn.predict(X_test_scaled)
knn_accuracy = accuracy_score(y_test, knn_pred)
print(f"KNN Accuracy: {knn_accuracy:.4f} ({knn_accuracy*100:.2f}%)")
```

```
KNN Accuracy: 0.8209 (82.09%)
```

In [13]:
```
print("KNN Classification Report:")
print(classification_report(y_test, knn_pred))
```

```
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.77      0.86       648
           1       0.64      0.94      0.76       279

    accuracy                           0.82       927
   macro avg       0.80      0.85      0.81       927
weighted avg       0.87      0.82      0.83       927
```

In [14]:
```
# 2. Support Vector Machine Classification
print("2. SUPPORT VECTOR MACHINE (SVM)")
svm = SVC(kernel='rbf', random_state=36)
svm.fit(X_train_scaled, y_train)
```

```
2. SUPPORT VECTOR MACHINE (SVM)
```

Out[14]:

SVC

▶ Parameters

In [15]:
```
svm_pred = svm.predict(X_test_scaled)
svm_accuracy = accuracy_score(y_test, svm_pred)
print(f"SVM Accuracy: {svm_accuracy:.4f} ({svm_accuracy*100}%)")
```

```
SVM Accuracy: 0.9461 (94.60625674217907%)
```

In [16]:
```
print("SVM Classification Report:")
print(classification_report(y_test, svm_pred))
```

```
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.99      0.96       648
           1       0.97      0.85      0.90       279

    accuracy                           0.95       927
   macro avg       0.95      0.92      0.93       927
weighted avg       0.95      0.95      0.94       927
```

In [17]:
```python
# Performance Analysis
print("=" * 50)
print("PERFORMANCE ANALYSIS")
print("=" * 50)
print(f"KNN Accuracy: {knn_accuracy*100}%")
print(f"SVM Accuracy: {svm_accuracy*100}%")
print("=" * 50)
```

```
==================================================
PERFORMANCE ANALYSIS
==================================================
KNN Accuracy: 82.09277238403452%
SVM Accuracy: 94.60625674217907%
==================================================
```

In [18]:
```python
if knn_accuracy > svm_accuracy:
    print(f"Best Model: KNN (Better by {(knn_accuracy-svm_accuracy)*100}%)")
else:
    print(f"Best Model: SVM (Better by {(svm_accuracy-knn_accuracy)*100}%)")
```

```
Best Model: SVM (Better by 12.513484358144556%)
```

## Group B

## Assignment No:3

**Title of the Assignment:** Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months

**Dataset Description:** The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

**Link for Dataset:** h ttps://www.kaggle.com/barelydedicated/bank-customer-churn-modeling

**Perform the following steps:**

1. Read the dataset.
2. Distinguish the feature and target set and divide the data set into training and test sets.
3. Normalize the train and test data.
4. Initialize and build the model. Identify the points of improvement and implement the same.
5. Print the accuracy score and confusion matrix (5 points).

**Objective of the Assignment:**

Students should be able to distinguish the feature and target set and divide the data set into training and test sets and normalize them and students should build the model on the basis of that.

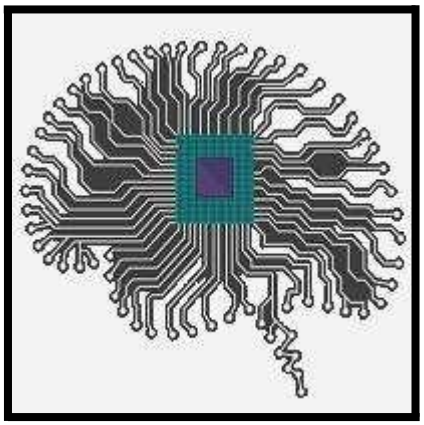
**Prerequisite:**

1. Basic knowledge of Python
2. Concept of Confusion Matrix

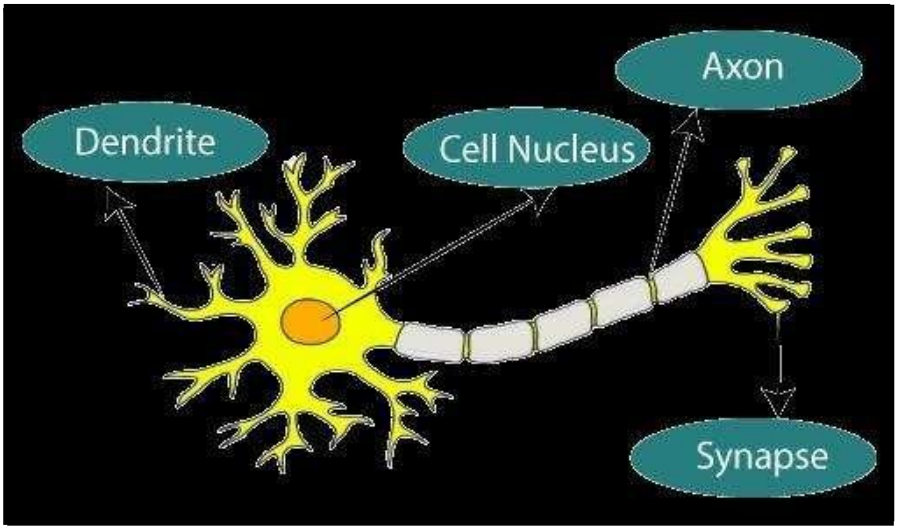
**Contents of the Theory:**

1. Artificial Neural Network
2. Keras
3. tensorflow
4. Normalization
5. Confusion Matrix

**Artificial Neural Network:**

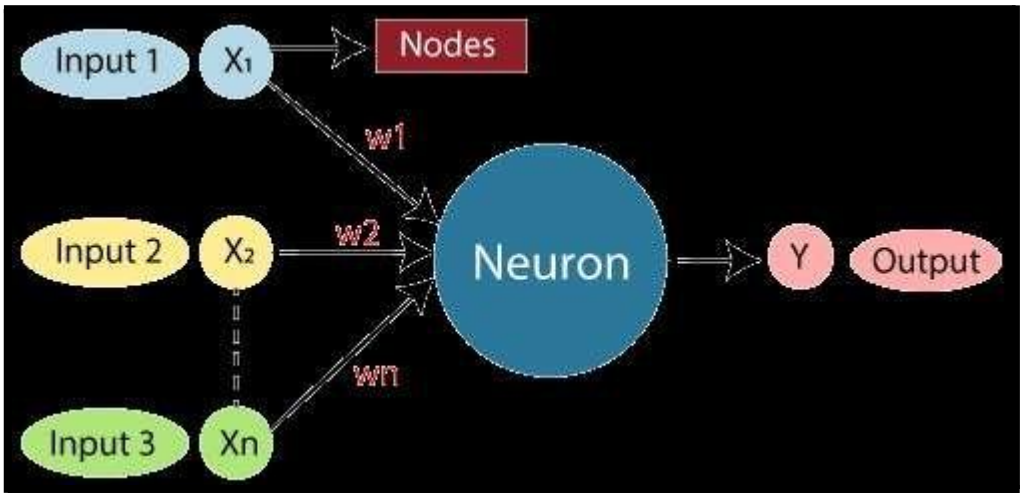


The term "Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.




The given figure illustrates the typical diagram of Biological Neural Network.

The typical Artificial Neural Network looks something like the given figure.




Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

Relationship between Biological neural network and artificial neural network:

| **Biological Neural Network** | **Artificial Neural Network** |
|---|---|
| Dendrites | Inputs |
| Cell nucleus | Nodes |
| Synapse | Weights |
| Axon | Output |

An **Artificial Neural Network** in the field of **Artificial intelligence** where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to

understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

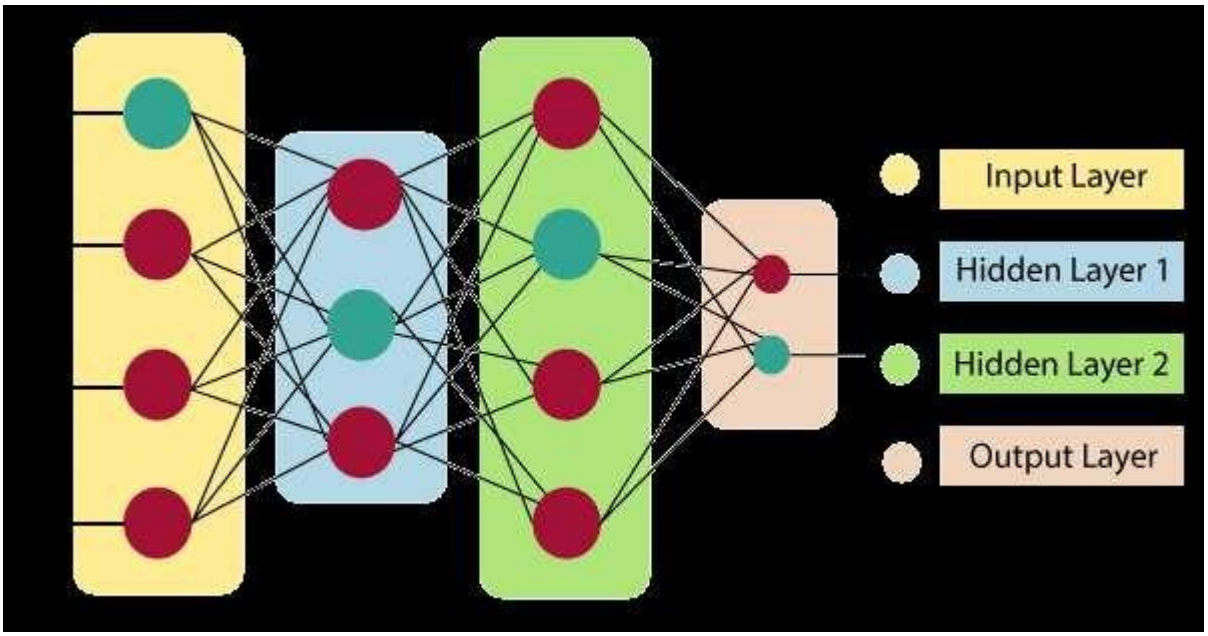
There are around 1000 billion neurons in the human brain. Each neuron has an association point somewhere in the range of 1,000 and 100,000. In the human brain, data is stored in such a manner as to be distributed, and we can extract more than one piece of this data when necessary from our memory parallelly. We can say that the human brain is made up of incredibly amazing parallel processors.

We can understand the artificial neural network with an example, consider an example of a digital logic gate that takes an input and gives an output. "OR" gate, which takes two inputs. If one or both the inputs are "On," then we get "On" in output. If both the inputs are "Off," then we get "Off" in output. Here the output depends upon input. Our brain does not perform the same task. The outputs to inputs relationship keep changing because of the neurons in our brain, which are "learning."

## The architecture of an artificial neural network:

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Lets us look at various types of layers available in an artificial neural network.

Artificial Neural Network primarily consists of three layers:




**Input Layer:**

As the name suggests, it accepts inputs in several different formats provided by the programmer.

**Hidden Layer:**

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

**Output Layer:**

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b$$

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.

**Keras:**

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user-friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

It cannot handle low-level computations, so it makes use of the**Backend**library to resolve it. The backend library act as a high-level API wrapper for the low-level API, which lets it run on TensorFlow, CNTK, or Theano.

Initially, it had over 4800 contributors during its launch, which now has gone up to 250,000 developers. It has a 2X growth ever since every year it has grown. Big companies like Microsoft, Google, NVIDIA, and Amazon have actively contributed to the development of Keras. It has an amazing industry interaction, and it is used in the development of popular firms likes Net flix, Uber, Google, Expedia, etc.


**Tensorflow:**

TensorFlow is a Google product, which is one of the most famous deep learning tools widely used in the research area of machine learning and deep neural network. It came into the market on 9th November 2015 under the Apache License 2.0. It is built in such a way that it can easily run on multiple CPUs and GPUs as well as on mobile operating systems. It consists of various wrappers in distinct languages such as Java, C++, or Python.


**Normalization:**

Normalization is a scaling technique in Machine Learning applied during data preparation to change the values of numeric columns in the dataset to use a common scale. It is not necessary for all datasets in a model. It is required only when features of machine learning models have different ranges.

Mathematically, we can calculate normalization with the below formula:

$$Xn = (X - Xminimum) / (Xmaximum - Xminimum)$$

Where,

- Xn = Value of Normalization

- Xmaximum = Maximum value of a feature
- Xminimum = Minimum value of a feature

**Example:** Let's assume we have a model dataset having maximum and minimum values of feature as mentioned above. To normalize the machine learning model, values are shifted and rescaled so their range can vary between 0 and 1. This technique is also known as Min-Max scaling. In this scaling technique, we will change the feature values as follows:

**Case1-**If the value of X is minimum, the value of Numerator will be 0; hence Normalization will also be 0.

Xn = (X - Xminimum) / ( Xmaximum - Xminimum)---------- formula

Put X =Xminimum in above formula, we get;

Xn = Xminimum- Xminimum/ ( Xmaximum - Xminimum)

Xn = 0

**Case2-**If the value of X is maximum, then the value of the numerator is equal to the denominator; hence Normalization will be 1.

Xn = (X - Xminimum) / ( Xmaximum - Xminimum)

Put X =Xmaximum in above formula, we get;

Xn = Xmaximum - Xminimum/ ( Xmaximum - Xminimum)

Xn = 1

**Case3-**On the other hand, if the value of X is neither maximum nor minimum, then values of normalization will also be between 0 and 1.

Hence, Normalization can be defined as a scaling method where values are shifted and rescaled to maintain their ranges between 0 and 1, or in other words; it can be referred to as Min-Max scaling technique.

## Normalization techniques in Machine Learning

Although there are so many feature normalization techniques in Machine Learning, few of them are most frequently used. These are as follows:

- **Min-Max Scaling:** This technique is also referred to as scaling. As we have already discussed above, the Min-Max scaling method helps the dataset to shift and rescale the values of their attributes, so they end up ranging between 0 and 1.
- **Standardization scaling:**

Standardization scaling is also known as **Z-score** normalization, in which values are centered around the mean with a unit standard deviation, which means the attribute becomes zero and the resultant distribution has a unit standard deviation. Mathematically, we can calculate the standardization by subtracting the feature value from the mean and dividing it by standard deviation.

Hence, standardization can be expressed as follows:

$$X' = \frac{X - \mu}{\sigma}$$

Here, **μ** represents the mean of feature value, and **σ** represents the standard deviation of feature values.

However, unlike Min-Max scaling technique, feature values are not restricted to a specific range in the standardization technique.

This technique is helpful for various machine learning algorithms that use distance measures such as **KNN, K-means clustering, and Principal component analysis**, etc. Further, it is also important that the model is built on assumptions and data is normally distributed.

## When to use Normalization or Standardization?

Which is suitable for our machine learning model, Normalization or Standardization? This is probably a big confusion among all data scientists as well as machine learning engineers. Although both terms have the almost same meaning choice of using normalization or standardization will depend on your problem and the algorithm you are using in models.

1. Normalization is a transformation technique that helps to improve the performance as well as the accuracy of your model better. Normalization of a machine learning model is useful when you don't know feature distribution exactly. In other words, the feature distribution of data does not follow a**Gaussian**(bell curve) distribution. Normalization must have an abounding range, so if you have outliers in data, they will be affected by Normalization.

Further, it is also useful for data having variable scaling techniques such as**KNN, arti ficial neural network**s. Hence, you can't use assumptions for the distribution of data.

2. Standardization in the machine learning model is useful when you are exactly aware of the feature distribution of data or, in other words, your data follows a Gaussian distribution. However, this does not have to be necessarily true. Unlike Normalization, Standardization does not necessarily have a bounding range, so if you have outliers in your data, they will not be affected by Standardization.

Further, it is also useful when data has variable dimensions and techniques such as**linear regression, logistic regression, and linear discriminant analysis**.

**Example:**Let's understand an experiment where we have a dataset having two attributes, i.e., age and salary. Where the age ranges from 0 to 80 years old, and the income varies from 0 to 75,000 dollars or more. Income is assumed to be 1,000 times that of age. As a result, the ranges of these two attributes are much different from one another.

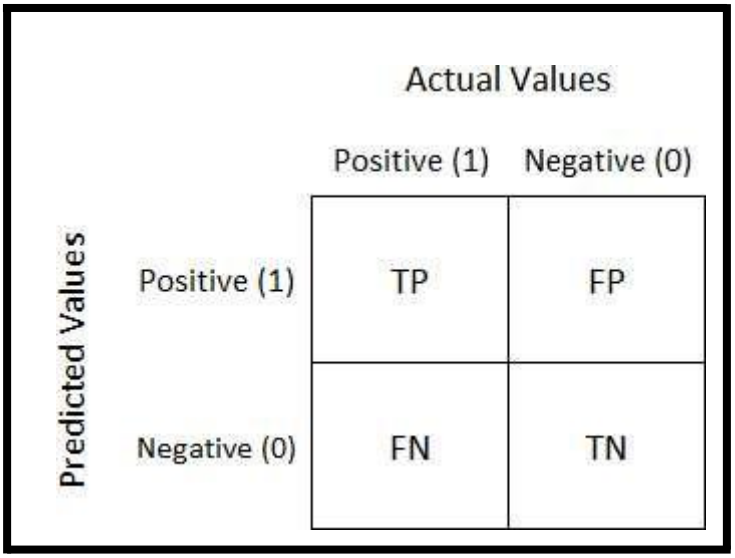
Because of its bigger value, the attributed income will organically in uence the conclusion more when we undertake further analysis, such as multivariate linear regression. However, this does not necessarily imply that it is a better predictor. As a result, we normalize the data so that all of the variables are in the same range.

Further, it is also helpful for the prediction of credit risk scores where normalization is applied to all numeric data except the class column. It uses the**tanh transformation** technique, which converts all numeric features into values of range between 0 to 1.

**Confusion Matrix:**

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an**error matrix**. Some features of Confusion matrix are given below:

- For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on.
- The matrix is divided into two dimensions, that are**predicted values**and**actual values**along with the total number of predictions.
- Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.
- It looks like the below table:



The above table has the following cases:

- **True Negative:** Model has given prediction No, and the real or actual value was also No.
- **True Positive:** The model has predicted yes, and the actual value was also true.
- **False Negative:** The model has predicted no, but the actual value was Yes, it is also called as **Type-II error**.
- **False Positive:** The model has predicted Yes, but the actual value was No. It is also called a **Type-I error.**

## Need for Confusion Matrix in Machine learning

- It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
- It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
- With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

**Example**: We can understand the confusion matrix using an example.

Suppose we are trying to create a model that can predict the result for the disease that is either a person has that disease or not. So, the confusion matrix for this is given as:

| n = 100 | Actual: No | Actual: Yes | |
|---|---|---|---|
| Predicted: No | TN: 65 | FP: 3 | 68 |
| Predicted: Yes | FN: 8 | TP: 24 | 32 |
| | 73 | 27 | |

From the above example, we can conclude that:

- The table is given for the two-class classifier, which has two predictions "Yes" and "NO." Here, Yes defines that patient has the disease, and No defines that patient does not has that disease.
- The classifier has made a total of **100 predictions**. Out of 100 predictions, **89 are true predictions**, and **11 are incorrect predictions**.
- The model has given prediction "yes" for 32 times, and "No" for 68 times. Whereas the actual "Yes" was 27, and actual "No" was 73 times.

Calculations using Confusion Matrix:

We can perform various calculations for the model, such as the model's accuracy, using this matrix. These calculations are given below:

- **Classification Accuracy:** It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculatedas the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

- **Misclassification rate:** It is also termed as Error rate, and it defines how often the model gives the wrong predictions. The value of error rate can be calculated as the number of incorrect predictions to all number of the predictions made by the classifier. The formula is given below:

$$\text{Error rate} = \frac{FP+FN}{TP+FP+FN+TN}$$

- **Precision:** It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:

$$\text{Precision} = \frac{TP}{TP+FP}$$

- **Recall:** It is defined as the out of total positive classes, how our model predicted correctly. The recall must be as high as possible.

$$\text{Recall} = \frac{TP}{TP+FN}$$

- **F-measure:** If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:

$$\text{F-measure} = \frac{2*Recall*Precision}{Recall+Precision}$$

Other important terms used in Confusion Matrix:

- **Null Error rate:** It defines how often our model would be incorrect if it always predicted the majority class. As per the accuracy paradox, it is said that "*the best classifier has a higher error rate than the null error rate.*"
- **ROC Curve:** The ROC is a graph displaying a classifier's performance for all possible thresholds. The graph is plotted between the true positive rate (on the Y-axis) and the false Positive rate (on the x-axis).

**Conclusion:**

In this way we build a a neural network-based classifier that can determine whether they will leave or not in the next 6 months

**Assignment Questions:**

1) What is Normalization?
2) What is Standardization?
3) Explain Confusion Matrix ?
4) Define the following: Classification Accuracy, Misclassification Rate, Precision.
5) One Example of Confusion Matrix?

In [24]:
```
# ------------------------------------------------------------------------------------------
# Step 1: Import required libraries
# ------------------------------------------------------------------------------------------
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.neural_network import MLPClassifier
```

In [25]:
```
# ------------------------------------------------------------------------------------------
# Step 2: Load the dataset and inspect its structure
# ------------------------------------------------------------------------------------------

# Path to dataset (update this if needed)
path = "churn_modelling.csv"

# Load CSV
df = pd.read_csv(path)

# Display basic info
print("Dataset shape:", df.shape)
print("\nColumns:", df.columns.tolist())
df.head()
```

```
Dataset shape: (10000, 14)

Columns: ['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography', 'Gender', 'Age',
'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary', 'Exite
d']
```

Out[25]:

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | Nu |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 15634602 | Hargrave | 619 | France | Female | 42 | 2 | 0.00 | |
| **1** | 2 | 15647311 | Hill | 608 | Spain | Female | 41 | 1 | 83807.86 | |
| **2** | 3 | 15619304 | Onio | 502 | France | Female | 42 | 8 | 159660.80 | |
| **3** | 4 | 15701354 | Boni | 699 | France | Female | 39 | 1 | 0.00 | |
| **4** | 5 | 15737888 | Mitchell | 850 | Spain | Female | 43 | 2 | 125510.82 | |

In [26]:
```
# ------------------------------------------------------------------------------------------
# Step 3: Prepare features (X) and target (y)
# ------------------------------------------------------------------------------------------

# Target variable
y = df['Exited']

# Drop identifiers and target from features
X = df.drop(columns=['RowNumber', 'CustomerId', 'Surname', 'Exited'], errors='ignore')

# Show feature sample
X.head()
```

Out[26]:

| | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMer |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 619 | France | Female | 42 | 2 | 0.00 | 1 | 1 | |
| **1** | 608 | Spain | Female | 41 | 1 | 83807.86 | 1 | 0 | |
| **2** | 502 | France | Female | 42 | 8 | 159660.80 | 3 | 1 | |
| **3** | 699 | France | Female | 39 | 1 | 0.00 | 2 | 0 | |
| **4** | 850 | Spain | Female | 43 | 2 | 125510.82 | 1 | 1 | |

In [27]:

```
# ------------------------------------------------------------------------------------
# Step 4: Encode categorical variables ('Gender', 'Geography')
# ------------------------------------------------------------------------------------

X_processed = X.copy()

# Encode Gender (Male = 1, Female = 0)
if 'Gender' in X_processed.columns:
    X_processed['Gender'] = X_processed['Gender'].map({'Male': 1, 'Female': 0})

# One-hot encode Geography
if 'Geography' in X_processed.columns:
    geo_dummies = pd.get_dummies(X_processed['Geography'], prefix='Geo', drop_first=True)
    X_processed = pd.concat([X_processed.drop(columns=['Geography']), geo_dummies], axis=1)

print("Processed feature columns:", X_processed.columns.tolist())
X_processed.head()
```

```
Processed feature columns: ['CreditScore', 'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProduct
s', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary', 'Geo_Germany', 'Geo_Spain']
```

Out[27]:

| | CreditScore | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | Estima |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 619 | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | |
| **1** | 608 | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | |
| **2** | 502 | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | |
| **3** | 699 | 0 | 39 | 1 | 0.00 | 2 | 0 | 0 | |
| **4** | 850 | 0 | 43 | 2 | 125510.82 | 1 | 1 | 1 | |

In [28]:

```
# ------------------------------------------------------------------------------------
# Step 5: Split dataset (80% train / 20% test)
# ------------------------------------------------------------------------------------

X_train, X_test, y_train, y_test = train_test_split(
    X_processed, y, test_size=0.2, random_state=42, stratify=y
)

print("Training set shape:", X_train.shape)
print("Testing set shape:", X_test.shape)
```

```
Training set shape: (8000, 11)
Testing set shape: (2000, 11)
```

In [29]:

```
# ------------------------------------------------------------------------------------
# Step 6: Standardize numeric features using StandardScaler
# ------------------------------------------------------------------------------------

scaler = StandardScaler()
```

```
num_cols = X_train.select_dtypes(include=[np.number]).columns.tolist()

# Fit on training data and transform both train & test
X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols] = scaler.transform(X_test[num_cols])

print("Mean of scaled training features (approx 0):")
print(X_train[num_cols].mean().round(3))
print("\nStandard deviation (approx 1):")
print(X_train[num_cols].std().round(3))
```

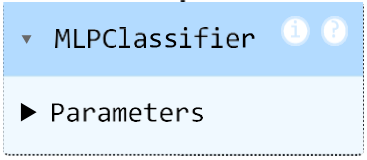
```
Mean of scaled training features (approx 0):
CreditScore       -0.0
Gender             0.0
Age                0.0
Tenure            -0.0
Balance            0.0
NumOfProducts     -0.0
HasCrCard          0.0
IsActiveMember     0.0
EstimatedSalary   -0.0
dtype: float64

Standard deviation (approx 1):
CreditScore        1.0
Gender             1.0
Age                1.0
Tenure             1.0
Balance            1.0
NumOfProducts      1.0
HasCrCard          1.0
IsActiveMember     1.0
EstimatedSalary    1.0
dtype: float64
```

In [30]:

```
# ------------------------------------------------------------------------------------------
# Step 7: Build Neural Network (MLP Classifier)
# ------------------------------------------------------------------------------------------

mlp = MLPClassifier(
    hidden_layer_sizes=(64, 32, 16),   # 3 hidden layers
    activation='relu',                 # ReLU activation
    solver='adam',                     # Adam optimizer
    alpha=1e-4,                        # L2 regularization term
    batch_size=64,                     # Mini-batch size
    learning_rate_init=0.001,          # Learning rate
    max_iter=200,                      # Max epochs
    early_stopping=True,               # Stop if no improvement
    validation_fraction=0.1,           # 10% validation split
    n_iter_no_change=10,               # Early stop patience
    random_state=42,
    verbose=True                       # Print training progress
)

# Train the model
mlp.fit(X_train, y_train)
```

```
Iteration 1, loss = 0.51298925
Validation score: 0.813750
Iteration 2, loss = 0.42146846
Validation score: 0.840000
Iteration 3, loss = 0.39445369
Validation score: 0.842500
Iteration 4, loss = 0.37218501
Validation score: 0.850000
Iteration 5, loss = 0.35436282
Validation score: 0.850000
Iteration 6, loss = 0.34469342
Validation score: 0.852500
Iteration 7, loss = 0.33831919
Validation score: 0.846250
Iteration 8, loss = 0.33639606
Validation score: 0.852500
Iteration 9, loss = 0.33105675
Validation score: 0.847500
Iteration 10, loss = 0.32876211
Validation score: 0.851250
Iteration 11, loss = 0.32503444
Validation score: 0.850000
Iteration 12, loss = 0.32284118
Validation score: 0.853750
Iteration 13, loss = 0.32041726
Validation score: 0.851250
Iteration 14, loss = 0.31701689
Validation score: 0.858750
Iteration 15, loss = 0.31600989
Validation score: 0.852500
Iteration 16, loss = 0.31428640
Validation score: 0.850000
Iteration 17, loss = 0.31019505
Validation score: 0.848750
Iteration 18, loss = 0.30916273
Validation score: 0.847500
Iteration 19, loss = 0.30748390
Validation score: 0.840000
Iteration 20, loss = 0.30614077
Validation score: 0.852500
Iteration 21, loss = 0.30352254
Validation score: 0.856250
Iteration 22, loss = 0.30193430
Validation score: 0.851250
Iteration 23, loss = 0.29919833
Validation score: 0.855000
Iteration 24, loss = 0.29818422
Validation score: 0.851250
Iteration 25, loss = 0.29599893
Validation score: 0.855000
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
```

Out[30]:




In [31]:

```
# ----------------------------------------------------------------------------------
# Step 8: Evaluate on test data
# ----------------------------------------------------------------------------------

y_pred = mlp.predict(X_test)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"\n✅ Test Accuracy: {accuracy:.4f}")
```

```python
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix (rows=True class, cols=Predicted class):")
print(cm)

# Detailed Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred, digits=4))
```

```
✅ Test Accuracy: 0.8565

Confusion Matrix (rows=True class, cols=Predicted class):
[[1534   59]
 [ 228  179]]

Classification Report:
              precision    recall  f1-score   support

           0     0.8706    0.9630    0.9145      1593
           1     0.7521    0.4398    0.5550       407

    accuracy                         0.8565      2000
   macro avg     0.8114    0.7014    0.7347      2000
weighted avg     0.8465    0.8565    0.8413      2000
```

# ML MINI PROJECT

**Problem Statement: -** Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.).

## Importing the Libraries

```
# linear algebra
import numpy as np

# data processing
import pandas as pd

# data visualization
import seaborn as sns
%matplotlib inline
from matplotlib import pyplot as plt
from matplotlib import style

# Algorithms
from sklearn import linear_model
from  sklearn.linear_model  import  LogisticRegression  from
sklearn.ensemble    import    RandomForestClassifier    from
sklearn.linear_model import Perceptron
from sklearn.linear_model import SGDClassifier from
sklearn.tree import DecisionTreeClassifier from
sklearn.neighbors import KNeighborsClassifier from
sklearn.svm import SVC, LinearSVC
from sklearn.naive_bayes import GaussianNB
```

## Getting the Data

```
test_df = pd.read_csv("test.csv")
train_df = pd.read_csv("train.csv")
```

## Data Exploration/Analysis

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId     891 non-null int64
Survived        891 non-null int64
Pclass          891 non-null int64
Name            891 non-null object
Sex             891 non-null object
Age             714 non-null float64
SibSp           891 non-null int64
Parch           891 non-null int64
Ticket          891 non-null object
Fare            891 non-null float64
Cabin           204 non-null object
Embarked        889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

**The training-set has 891 examples and 11 features + the target variable (survived)**. 2 of the features are floats, 5 are integers and 5 are objects. Below I have listed the features with a short description:

```
survival:     Survival
PassengerId: Unique Id of a passenger.
pclass:    Ticket class
sex:    Sex
Age:    Age in years
sibsp: # of siblings / spouses aboard the Titanic parch:
# of parents / children aboard the Titanic ticket:
Ticket number
fare:    Passenger fare
cabin:    Cabin number
embarked:    Port of Embarkationtrain df.describe()
```

| | PassengerId | Survived | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|---|
| count | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000000 |
| mean | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204208 |
| std | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 | 49.693429 |
| min | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 | 7.910400 |
| 50% | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454200 |
| 75% | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000000 |
| max | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329200 |

Above we can see that **38% out of the training-set survived the Titanic**. We can also see that the passenger ages range from 0.4 to 80. On top of that we can already detect some features, that contain missing values, like the „Age" feature.

```
train_df.head(8)
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| 1 | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 2 | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |
| 3 | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| 4 | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.0500 | NaN | S |
| 5 | 6 | 0 | 3 | Moran, Mr. James | male | NaN | 0 | 0 | 330877 | 8.4583 | NaN | Q |
| 6 | 7 | 0 | 1 | McCarthy, Mr. Timothy J | male | 54.0 | 0 | 0 | 17463 | 51.8625 | E46 | S |
| 7 | 8 | 0 | 3 | Palsson, Master. Gosta Leonard | male | 2.0 | 3 | 1 | 349909 | 21.0750 | NaN | S |
| 8 | 9 | 1 | 3 | Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) | female | 27.0 | 0 | 2 | 347742 | 11.1333 | NaN | S |

From the table above, we can note a few things. First of all, that we **need to convert a lot of features into numeric** ones later on, so that the machine learning algorithms can process them. Furthermore, we can see that the **features have widely different ranges**, that we will need to convert into roughly the same scale. We can also spot some more features, that contain missing values (NaN = not a number), that wee need to deal with.

### Let's take a more detailed look at what data is actually missing:

```
total = train_df.isnull().sum().sort_values(ascending=False)
percent_1 = train_df.isnull().sum()/train_df.isnull().count()*100
percent_2 = (round(percent_1, 1)).sort_values(ascending=False)
missing_data = pd.concat([total, percent_2], axis=1, keys=['Total',
'%'])
missing_data.head(5)
```

| | Total | % |
|---|---|---|
| Cabin | 687 | 77.1 |
| Age | 177 | 19.9 |
| Embarked | 2 | 0.2 |
| Fare | 0 | 0.0 |
| Ticket | 0 | 0.0 |

The Embarked feature has only 2 missing values, which can easily be filled. It will be much more tricky, to deal with the „Age" feature, which has 177 missing values. The „Cabin" feature needs further investigation, but it looks like that we might want to drop it from the dataset, since 77 % of it are missing.

```
train_df.columns.values
```

```
array(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'], dtype=object)
```

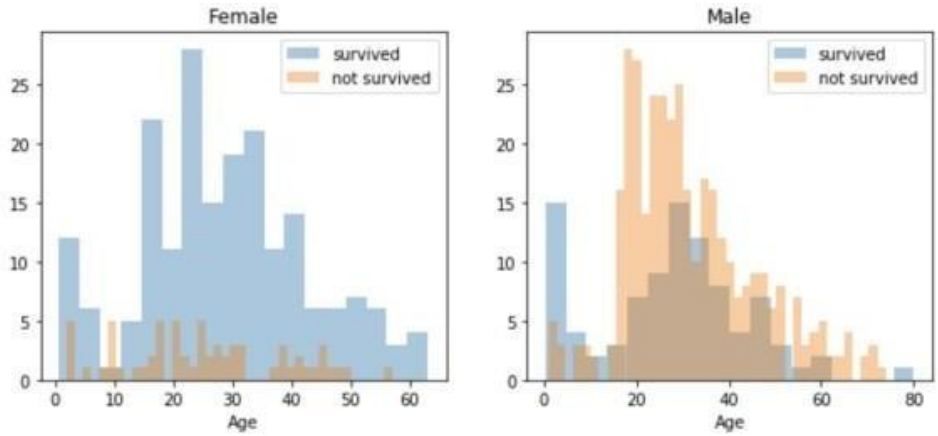
Above you can see the 11 features + the target variable (survived). **What features could contribute to a high survival rate ?**

To me it would make sense if everything except „PassengerId", „Ticket" and „Name" would be correlated with a high survival rate.

### 1. Age and Sex:

```
survived = 'survived'
not_survived = 'not survived'
fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(10, 4))
women = train_df[train_df['Sex']=='female']
men = train_df[train_df['Sex']=='male']
ax = sns.distplot(women[women['Survived']==1].Age.dropna(), bins=18,
label = survived, ax = axes[0], kde =False)
ax = sns.distplot(women[women['Survived']==0].Age.dropna(), bins=40,
label = not_survived, ax = axes[0], kde =False)
ax.legend()
ax.set_title('Female')
ax = sns.distplot(men[men['Survived']==1].Age.dropna(), bins=18, label
= survived, ax = axes[1], kde = False)
```

```
ax = sns.distplot(men[men['Survived']==0].Age.dropna(), bins=40, label
= not_survived, ax = axes[1], kde = False)
ax.legend()
  = ax.set title('Male')
```




You can see that men have a high probability of survival when they are between 18 and 30 years old, which is also a little bit true for women but not fully. For women the survival chances are higher between 14 and 40.

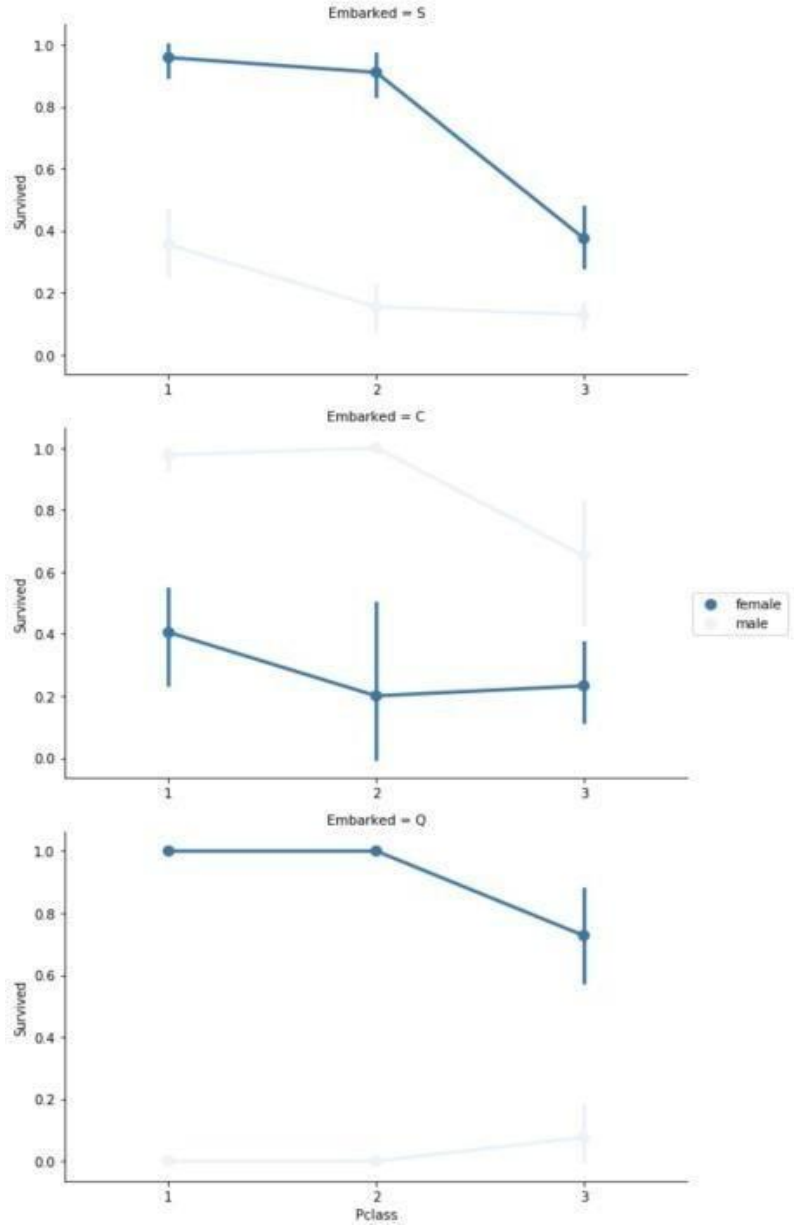
For men the probability of survival is very low between the age of 5 and 18, but that isn" t true for women. Another thing to note is that infants also have a little bit higher probability of survival.

Since there seem to be **certain ages, which have increased odds of survival** and because I want every feature to be roughly on the same scale, I will create age groups later on.

### 3. Embarked, Pclass and Sex:

```
FacetGrid = sns.FacetGrid(train_df, row='Embarked', size=4.5,
aspect=1.6)
FacetGrid.map(sns.pointplot, 'Pclass', 'Survived', 'Sex',
palette=None, order=None, hue_order=None )
FacetGrid.add_legend()
```

```
<seaborn.axisgrid.FacetGrid at 0x10ba485c0>
```




Embarked seems to be correlated with survival, depending on the gender.

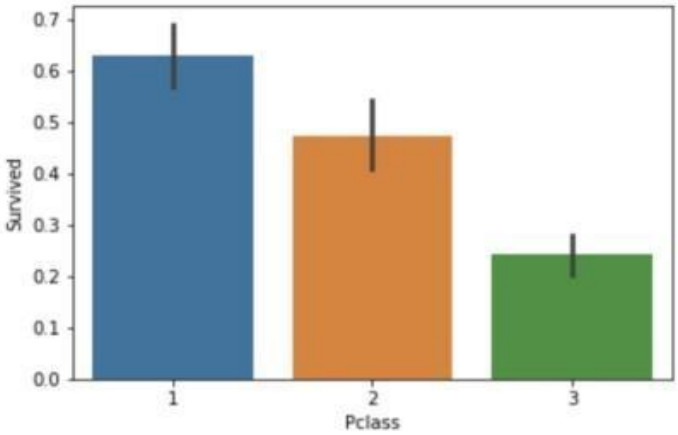
Women on port Q and on port S have a higher chance of survival. The inverse is true, if they are at port C. Men have a high survival probability if they are on port C, but a low probability if they are on port Q or S.

Pclass also seems to be correlated with survival. We will generate another plot of it below.

### 4. Pclass:

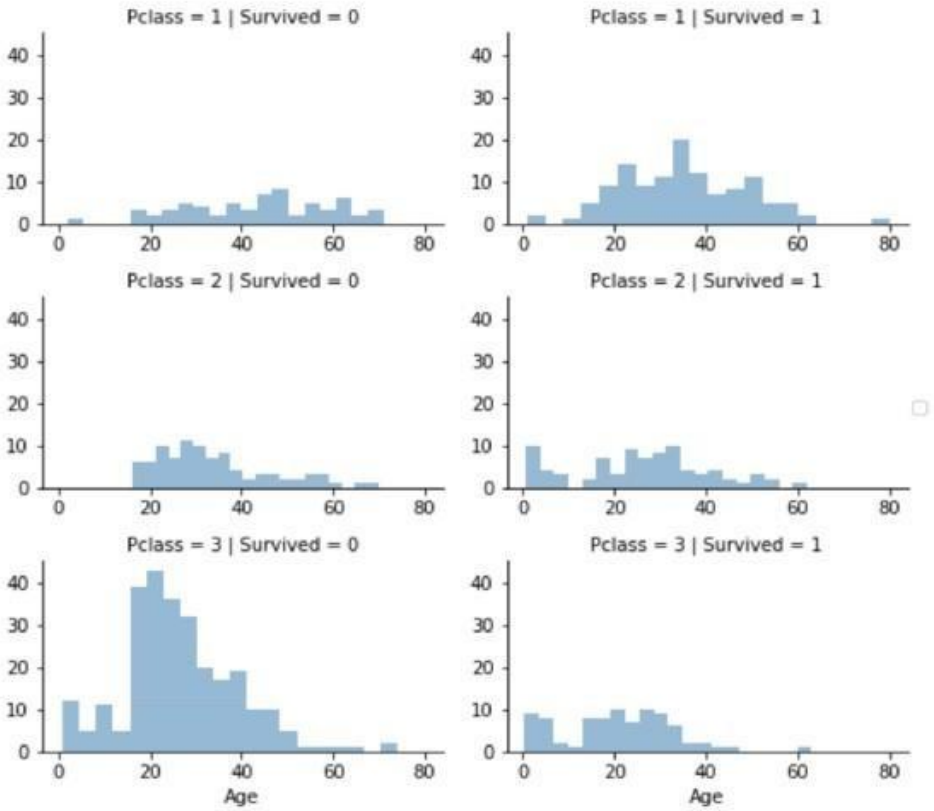
```
sns.barplot(x='Pclass', y='Survived', data=train_df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x10d1dc7b8>
```




Here we see clearly, that Pclass is contributing to a persons chance of survival, especially if this person is in class 1. We will create another pclass plot below.

```
grid = sns.FacetGrid(train_df, col='Survived', row='Pclass',
size=2.2, aspect=1.6)
grid.map(plt.hist, 'Age', alpha=.5, bins=20)
grid.add_legend();
```




The plot above confirms our assumption about pclass 1, but we can also spot a high probability that a person in pclass 3 will not survive.

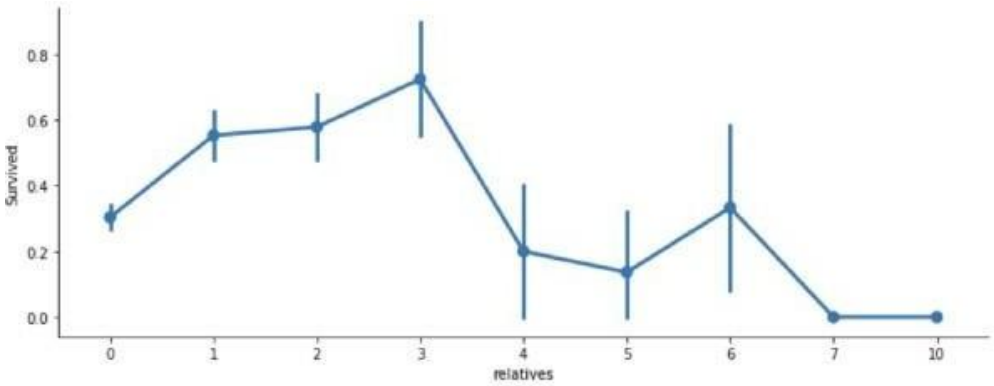
**5. SibSp and Parch:**

SibSp and Parch would make more sense as a combined feature, that shows the total number of relatives, a person has on the Titanic. I will create it below and also a feature that sows if so

```
data = [train_df, test_df]
for dataset in data:
    dataset['relatives'] = dataset['SibSp'] + dataset['Parch']
    dataset.loc[dataset['relatives'] > 0, 'not_alone'] = 0
    dataset.loc[dataset['relatives'] == 0, 'not_alone'] = 1
    dataset['not_alone'] =
dataset['not alone'].astype(int)train df['not alone'].value coun
```

```
1    537
0    354
Name: not_alone, dtype: int64
```

```
axes = sns.factorplot('relatives','Survived',
                      data=train df, aspect = 2.5, )
```




Here we can see that you had a high probabilty of survival with 1 to 3 realitves, but a lower one if you had less than 1 or more than 3 (except for some cases with 6 relatives).

## Data Preprocessing

First, I will drop „PassengerId" from the train set, because it does not contribute to a persons survival probability. I will not drop it from the test set, since it is required there for the submission.

```
train_df = train_df.drop(['PassengerId'], axis=1)
```

Missing Data:

**Cabin:**

As a reminder, we have to deal with Cabin (687), Embarked (2) and Age (177). First I thought, we have to delete the „Cabin" variable but then I

found something interesting. A cabin number looks like „C123" and the **letter refers to the deck**. Therefore we're going to extract these and create a new feature, that contains a persons deck. Afterwords we will convert the feature into a numeric variable. The missing values will be converted to zero. In the picture below you can see the actual decks of the titanic, ranging from A to G.

```
import re
deck = {"A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6, "G": 7, "U": 8}
data = [train_df, test_df]

for dataset in data:
    dataset['Cabin'] = dataset['Cabin'].fillna("U0")
    dataset['Deck'] = dataset['Cabin'].map(lambda x: re.compile("([a-zA-Z]+)").search(x).group())
    dataset['Deck'] = dataset['Deck'].map(deck)
    dataset['Deck'] = dataset['Deck'].fillna(0)
    dataset['Deck'] = dataset['Deck'].astype(int)# we can now drop the cabin feature
train_df = train_df.drop(['Cabin'], axis=1)
test_df = test_df.drop(['Cabin'], axis=1)
```

### Age:

Now we can tackle the issue with the age features missing values. I will create an array that contains random numbers, which are computed based on the mean age value in regards to the standard deviation and is_null.

```
data = [train_df, test_df]

for dataset in data:
    mean = train_df["Age"].mean()
    std = test_df["Age"].std()
    is_null = dataset["Age"].isnull().sum()
    # compute random numbers between the mean, std and is_null
    rand_age = np.random.randint(mean - std, mean + std, size = is_null)
    # fill NaN values in Age column with random values generated
    age_slice = dataset["Age"].copy()
    age_slice[np.isnan(age_slice)] = rand_age
    dataset["Age"] = age_slice
    dataset["Age"] = train df["Age"].astype(int)train df["Age"].isnull().sum()
```

### Embarked:

Since the Embarked feature has only 2 missing values, we will just fill these with the most common one.

```
train_df['Embarked'].describe()
```

```
count      889
unique       3
top          S
freq       644
Name: Embarked, dtype: object
```

```
common_value = 'S'
data = [train_df, test_df]

for dataset in data:
    dataset['Embarked'] = dataset['Embarked'].fillna(common_value)
```

Converting Features:

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 13 columns):
Survived     891 non-null int64
Pclass       891 non-null int64
Name         891 non-null object
Sex          891 non-null object
Age          891 non-null int64
SibSp        891 non-null int64
Parch        891 non-null int64
Ticket       891 non-null object
Fare         891 non-null float64
Embarked     891 non-null object
relatives    891 non-null int64
not_alone    891 non-null int64
Deck         891 non-null int64
dtypes: float64(1), int64(8), object(4)
memory usage: 90.6+ KB
```

Above you can see that „Fare" is a float and we have to deal with 4 categorical features: Name, Sex, Ticket and Embarked. Lets investigate and transfrom one after another.

**Fare:**

Converting "Fare" from float to int64, using the "astype()" function pandas provides:

```
data = [train_df, test_df]

for dataset in data:
    dataset['Fare'] = dataset['Fare'].fillna(0)
    dataset['Fare'] = dataset['Fare'].astype(int)
```

**Name:**

We will use the Name feature to extract the Titles from the Name, so that we can build a new feature out of that.

```
data = [train_df, test_df]
titles = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Rare": 5}

for dataset in data:
    # extract titles
```

```
    dataset['Title'] = dataset.Name.str.extract(' ([A-Za-z]+)\.',
expand=False)
    # replace titles with a more common title or as Rare
    dataset['Title'] = dataset['Title'].replace(['Lady',
'Countess','Capt', 'Col','Don', 'Dr',\
                                            'Major', 'Rev', 'Sir',
'Jonkheer', 'Dona'], 'Rare')
    dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Ms', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')
    # convert titles into numbers
    dataset['Title'] = dataset['Title'].map(titles)
    # filling NaN with 0, to get safe
    dataset['Title'] = dataset['Title'].fillna(0)train_df =
train_df.drop(['Name'], axis=1)
test df = test df.drop(['Name'], axis=1)
```

### Sex:

Convert „Sex" feature into numeric.

```
genders = {"male": 0, "female": 1}
data = [train_df, test_df]

for dataset in data:
    dataset['Sex'] = dataset['Sex'].map(genders)
```

### Ticket:

```
train df['Ticket'].describe()
```

```
count      891
unique     681
top       1601
freq         7
Name: Ticket, dtype: object
```

Since the Ticket attribute has 681 unique tickets, it will be a bit tricky to convert them into useful categories. So we will drop it from the dataset.

```
train_df = train_df.drop(['Ticket'], axis=1)
test_df = test_df.drop(['Ticket'], axis=1)
```

### Embarked:

Convert „Embarked" feature into numeric.

```
ports = {"S": 0, "C": 1, "Q": 2}
data = [train_df, test_df]

for dataset in data:
    dataset['Embarked'] = dataset['Embarked'].map(ports)
```

### Creating Categories:

We will now create categories within the following features:

**Age:**

Now we need to convert the „age" feature. First we will convert it from float into integer. Then we will create the new „AgeGroup" variable, by categorizing every age into a group. Note that it is important to place attention on how you form these groups, since you don" t want for example that 80% of your data falls into group 1.

```
data = [train_df, test_df]
for dataset in data:
    dataset['Age'] = dataset['Age'].astype(int)
    dataset.loc[ dataset['Age'] <= 11, 'Age'] = 0
    dataset.loc[(dataset['Age'] > 11) & (dataset['Age'] <= 18), 'Age']
= 1
    dataset.loc[(dataset['Age'] > 18) & (dataset['Age'] <= 22), 'Age']
= 2
    dataset.loc[(dataset['Age'] > 22) & (dataset['Age'] <= 27), 'Age']
= 3
    dataset.loc[(dataset['Age'] > 27) & (dataset['Age'] <= 33), 'Age']
= 4
    dataset.loc[(dataset['Age'] > 33) & (dataset['Age'] <= 40), 'Age']
= 5
    dataset.loc[(dataset['Age'] > 40) & (dataset['Age'] <= 66), 'Age']
= 6
    dataset.loc[ dataset['Age'] > 66, 'Age'] = 6

# let's see how it's distributed train_df['Age'].value_counts()
```

```
4    165
6    158
5    147
3    129
2    124
1    100
0     68
Name: Age, dtype: int64
```

**Fare:**

For the „Fare" feature, we need to do the same as with the „Age" feature. But it isn" t that easy, because if we cut the range of the fare values into a few equally big categories, 80% of the values would fall into the first category. Fortunately, we can use sklearn "qcut()" function, that we can use to see, how we can form the categories.

```
train_df.head(10)
```

| | Survived | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked | relatives | not_alone | Deck | Title |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | 0 | 2 | 1 | 0 | 7 | 0 | 1 | 0 | 8 | 1 |
| **1** | 1 | 1 | 1 | 5 | 1 | 0 | 71 | 1 | 1 | 0 | 3 | 3 |
| **2** | 1 | 3 | 1 | 3 | 0 | 0 | 7 | 0 | 0 | 1 | 8 | 2 |
| **3** | 1 | 1 | 1 | 5 | 1 | 0 | 53 | 0 | 1 | 0 | 3 | 3 |
| **4** | 0 | 3 | 0 | 5 | 0 | 0 | 8 | 0 | 0 | 1 | 8 | 1 |
| **5** | 0 | 3 | 0 | 4 | 0 | 0 | 8 | 2 | 0 | 1 | 8 | 1 |
| **6** | 0 | 1 | 0 | 6 | 0 | 0 | 51 | 0 | 0 | 1 | 5 | 1 |
| **7** | 0 | 3 | 0 | 0 | 3 | 1 | 21 | 0 | 4 | 0 | 8 | 4 |
| **8** | 1 | 3 | 1 | 3 | 0 | 2 | 11 | 0 | 2 | 0 | 8 | 3 |
| **9** | 1 | 2 | 1 | 1 | 1 | 0 | 30 | 1 | 1 | 0 | 8 | 3 |

```
data = [train_df, test_df]

for dataset in data:
    dataset.loc[ dataset['Fare'] <= 7.91, 'Fare'] = 0
    dataset.loc[(dataset['Fare'] > 7.91) & (dataset['Fare'] <=
14.454), 'Fare'] = 1
    dataset.loc[(dataset['Fare'] > 14.454) & (dataset['Fare'] <= 31),
'Fare']   = 2
    dataset.loc[(dataset['Fare'] > 31) & (dataset['Fare'] <= 99),
'Fare']   = 3
    dataset.loc[(dataset['Fare'] > 99) & (dataset['Fare'] <= 250),
'Fare']   = 4
    dataset.loc[ dataset['Fare'] > 250, 'Fare'] = 5
    dataset['Fare'] = dataset['Fare'].astype(int)
```

## Creating new Features

I will add two new features to the dataset, that I compute out of other features.

### 1. Age times Class

```
data = [train_df, test_df]
for dataset in data:
    dataset['Age_Class']= dataset['Age']* dataset['Pclass']
```

### 2. Fare per Person

```
for dataset in data:
    dataset['Fare_Per_Person'] =
dataset['Fare']/(dataset['relatives']+1)
    dataset['Fare_Per_Person'] =
dataset['Fare_Per_Person'].astype(int)# Let's take a last look at the
training set, before we start training the models.
train df.head(10)
```

| | Survived | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked | relatives | not_alone | Deck | Title | Age_Class | Fare_Per_Pers |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 8 | 1 | 6 | 0 |
| **1** | 1 | 1 | 1 | 5 | 1 | 0 | 3 | 1 | 1 | 0 | 3 | 3 | 5 | 1 |
| **2** | 1 | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 8 | 2 | 9 | 0 |
| **3** | 1 | 1 | 1 | 5 | 1 | 0 | 3 | 0 | 1 | 0 | 3 | 3 | 5 | 1 |
| **4** | 0 | 3 | 0 | 5 | 0 | 0 | 1 | 0 | 0 | 1 | 8 | 1 | 15 | 1 |
| **5** | 0 | 3 | 0 | 4 | 0 | 0 | 1 | 2 | 0 | 1 | 8 | 1 | 12 | 1 |
| **6** | 0 | 1 | 0 | 6 | 0 | 0 | 3 | 0 | 0 | 1 | 5 | 1 | 6 | 3 |
| **7** | 0 | 3 | 0 | 0 | 3 | 1 | 2 | 0 | 4 | 0 | 8 | 4 | 0 | 0 |
| **8** | 1 | 3 | 1 | 3 | 0 | 2 | 1 | 0 | 2 | 0 | 8 | 3 | 9 | 0 |
| **9** | 1 | 2 | 1 | 1 | 1 | 0 | 2 | 1 | 1 | 0 | 8 | 3 | 2 | 1 |
| **10** | 1 | 3 | 1 | 0 | 1 | 1 | 2 | 0 | 2 | 0 | 7 | 2 | 0 | 0 |

## Building Machine Learning Models

Now we will train several Machine Learning models and compare their results. Note that because the dataset does not provide labels for their testing-set, we need to use the predictions on the training set to compare the algorithms with each other. Later on, we will use cross validation.

```
X_train = train_df.drop("Survived", axis=1)
Y_train = train_df["Survived"]
X_test  = test_df.drop("PassengerId", axis=1).copy()
```

### Stochastic Gradient Descent (SGD):

```
sgd = linear_model.SGDClassifier(max_iter=5, tol=None)
sgd.fit(X_train, Y_train)
Y_pred = sgd.predict(X_test)

sgd.score(X_train, Y_train)

acc_sgd = round(sgd.score(X_train, Y_train) * 100, 2)
```

### Random Forest:

```
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, Y_train)

Y_prediction = random_forest.predict(X_test)

random_forest.score(X_train, Y_train)
acc_random_forest = round(random_forest.score(X_train, Y_train) * 100,
2)
```

### Logistic Regression:

```
logreg = LogisticRegression()
logreg.fit(X_train, Y_train)

Y_pred = logreg.predict(X_test)

acc_log = round(logreg.score(X_train, Y_train) * 100, 2)
```

## K Nearest Neighbor:

```
# KNN knn = KNeighborsClassifier(n_neighbors = 3) knn.fit(X_train,
Y_train) Y_pred = knn.predict(X_test) acc_knn =
round(knn.score(X_train, Y_train) * 100, 2)
```

## Gaussian Naive Bayes:

```
gaussian = GaussianNB() gaussian.fit(X_train, Y_train) Y_pred =
gaussian.predict(X_test) acc_gaussian = round(gaussian.score(X_train,
Y_train) * 100, 2)
```

## Perceptron:

```
perceptron = Perceptron(max_iter=5)
perceptron.fit(X_train, Y_train)

Y_pred = perceptron.predict(X_test)

acc perceptron = round(perceptron.score(X train, Y train) * 100, 2)
```

## Linear Support Vector Machine:

```
linear_svc = LinearSVC()
linear_svc.fit(X_train, Y_train)

Y_pred = linear_svc.predict(X_test)

acc_linear_svc = round(linear_svc.score(X_train, Y_train) * 100, 2)
```

## Decision Tree

```
decision_tree = DecisionTreeClassifier() decision_tree.fit(X_train,
Y_train) Y_pred = decision_tree.predict(X_test)  acc_decision_tree =
round(decision_tree.score(X_train, Y_train) * 100, 2)
```

## Which is the best Model ?

```
results = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression',
              'Random Forest', 'Naive Bayes', 'Perceptron',
              'Stochastic Gradient Decent',
              'Decision Tree'],
    'Score': [acc_linear_svc, acc_knn, acc_log,
              acc_random_forest, acc_gaussian, acc_perceptron,
              acc_sgd, acc_decision_tree]})
result_df = results.sort_values(by='Score', ascending=False)
result_df = result_df.set_index('Score')
result df.head(9)
```

| | Model |
|---|---|
| Score | |
| 92.82 | Random Forest |
| 92.82 | Decision Tree |
| 87.32 | KNN |
| 81.14 | Logistic Regression |
| 80.81 | Support Vector Machines |
| 80.70 | Perceptron |
| 77.10 | Naive Bayes |
| 76.99 | Stochastic Gradient Decent |

As we can see, the Random Forest classifier goes on the first place. But first, let us check, how random-forest performs, when we use cross validation.

## K-Fold Cross Validation:

K-Fold Cross Validation randomly splits the training data into **K subsets called folds**. Let' s image we would split our data into 4 folds (K = 4). Our random forest model would be trained and evaluated 4 times, using a different fold for evaluation everytime, while it would be trained on the remaining 3 folds.

The image below shows the process, using 4 folds (K = 4). Every row represents one training + evaluation process. In the first row, the model get" s trained on the first, second and third subset and evaluated on the fourth. In the second row, the model get" s trained on the second, third and fourth subset and evaluated on the first. K-Fold Cross Validation repeats this process till every fold acted once as an evaluation fold.

| Training | Training | Training | Evaluation |
|---|---|---|---|
| 1 | 2 | 3 | **4** |
| 2 | 3 | 4 | **1** |
| 3 | 4 | 1 | **2** |
| 4 | 1 | 2 | **3** |

The result of our K-Fold Cross Validation example would be an array that contains 4 different scores. We then need to compute the mean and the standard deviation for these scores.

The code below perform K-Fold Cross Validation on our random forest model, using 10 folds (K = 10). Therefore it outputs an array with 10 different scores.

```
from sklearn.model_selection import
cross_val_score rf =
RandomForestClassifier(n_estimators=100)
scores = cross_val_score(rf, X_train, Y_train, cv=10, scoring =
"accuracy")print("Scores:", scores)
print("Mean:", scores.mean())
```

```
Scores: [ 0.76666667  0.82222222  0.7752809   0.82022472  0.85393258  0.86516854
  0.83146067  0.76404494  0.85393258  0.85227273]
Mean: 0.820520655998
Standard Deviation: 0.0367333665466
```

This looks much more realistic than before. Our model has a average accuracy of 82% with a standard deviation of 4 %. The standard deviation shows us, how precise the estimates are .

This means in our case that the accuracy of our model can differ + — 4%.

I think the accuracy is still really good and since random forest is an easy to use model, we will try to increase it" s performance even further in the following section.

## Random Forest

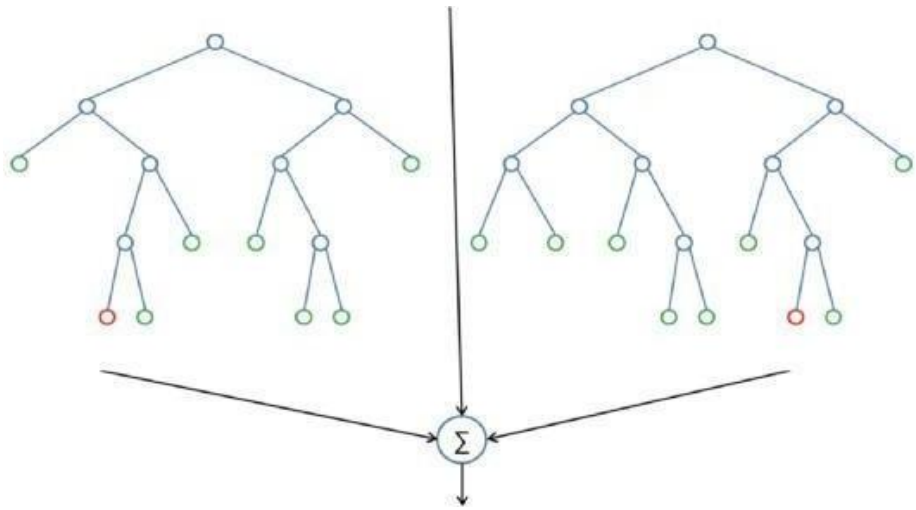
### What is Random Forest ?

Random Forest is a supervised learning algorithm. Like you can already see from it" s name, it creates a forest and makes it somehow random. The „forest“ it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.
To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. With a few exceptions a random-forest classifier has all the hyperparameters of a decision-tree classifier and also all the hyperparameters of a bagging classifier, to control the ensemble itself.

The random-forest algorithm brings extra randomness into the model, when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better model. Therefore when you are growing a tree in random forest, only a random subset of the features is considered for splitting a node. You can even make trees more random, by using random thresholds on top of it, for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Below you can see how a random forest would look like with two trees:




## Feature Importance

Another great quality of random forest is that they make it very easy to measure the relative importance of each feature. Sklearn measure a features importance by looking at how much the treee

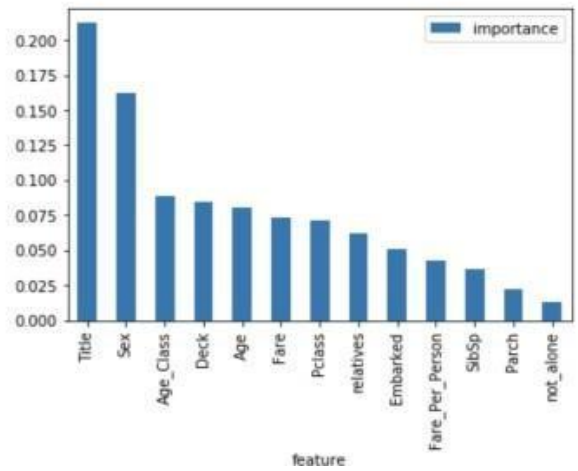
feature, reduce impurity on average (across all trees in the forest). It computes this score automaticall for each feature after training and scales the results so that the sum of all importances is equal to 1. We will acces this below:

```
importances =
pd.DataFrame({'feature':X_train.columns,'importance':np.round(rand
om_f orest.feature_importances_,3)})
importances =
importances.sort_values('importance',ascending=False).set_index('f
eatu re')importances.head(15)
```

| | importance |
|---|---|
| feature | |
| Title | 0.212 |
| Sex | 0.162 |
| Age_Class | 0.089 |
| Deck | 0.084 |
| Age | 0.080 |
| Fare | 0.073 |
| Pclass | 0.071 |
| relatives | 0.062 |
| Embarked | 0.051 |
| Fare_Per_Person | 0.043 |
| SibSp | 0.036 |
| Parch | 0.022 |
| not_alone | 0.013 |

```
importances.plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a157c1e10>
```




Conclusion:

not_alone and Parch doesn" t play a significant role in our random forest classifiers prediction process. Because of that I will drop them from the dataset and train the classifier again. We could also remove more or less

features, but this would need a more detailed investigation of the features effect on our model. But I think it" s just fine to remove only Alone and Parch.

```
train_df = train_df.drop("not_alone",
axis=1) test_df =
test_df.drop("not_alone", axis=1)

train_df = train_df.drop("Parch",
```

**Training random forest again:**

```
# Random Forest

random_forest = RandomForestClassifier(n_estimators=100, oob_score =
True)
random_forest.fit(X_train, Y_train)
Y_prediction = random_forest.predict(X_test)

random_forest.score(X_train, Y_train)

acc_random_forest = round(random_forest.score(X_train, Y_train) * 100,
2)
print(round(acc_random_forest,2,), "%")
```

92.82%

Our random forest model predicts as good as it did before. A general rule is that, **the more features you have, the more likely your model will suffer from overfitting** and vice versa. But I think our data looks fine for now and hasn't too much features.

There is also another way to evaluate a random-forest classifier, which is probably much more accurate than the score we used before. What I am talking about is the **out-of-bag samples** to estimate the generalization accuracy. I will not go into details here about how it works. Just note that out-of-bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out-of-bag error estimate removes the need for a set aside test set.

```
print("oob score:", round(random_forest.oob_score_, 4)*100,
"%")
```

oob score: 81.82 %

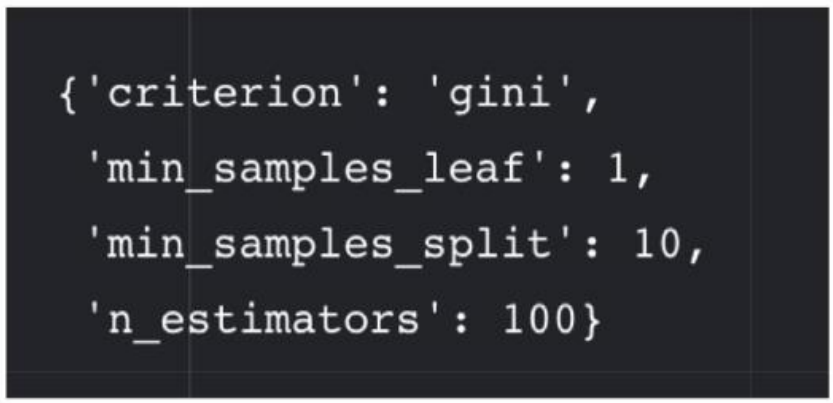
Now we can start tuning the hyperameters of random forest.

## Hyperparameter Tuning

Below you can see the code of the hyperparamter tuning for the parameters criterion, min_samples_leaf, min_samples_split and n_estimators.

I put this code into a markdown cell and not into a code cell, because it takes a long time to run it. Directly underneeth it, I put a screenshot of the gridsearch's output.

```
param_grid = { "criterion" : ["gini", "entropy"],
"min_samples_leaf" : [1, 5, 10, 25, 50, 70], "min_samples_split"
: [2, 4, 10, 12, 16, 18,
25, 35], "n_estimators": [100, 400, 700, 1000, 1500]}from
sklearn.model_selection import GridSearchCV, cross_val_scorerf =
RandomForestClassifier(n_estimators=100, max_features='auto',
oob_score=True, random_state=1, n_jobs=-1)clf =
GridSearchCV(estimator=rf, param_grid=param_grid, n_jobs=-
```




**Test new Parameters:**

```
# Random Forest
random_forest = RandomForestClassifier(criterion = "gini",
                                       min_samples_leaf = 1,
                                       min_samples_split = 10,
                                       n_estimators=100,
                                       max_features='auto',
                                       oob_score=True,
                                       random_state=1, n_jobs=-1)

random_forest.fit(X_train, Y_train)
Y_prediction = random_forest.predict(X_test)

random_forest.score(X_train, Y_train)

print("oob score:", round(random_forest.oob_score_, 4)*100, "%")
```

oob score: 83.05 %

Now that we have a proper model, we can start evaluating it" s performace in a more accurate way. Previously we only used accuracy and the oob score, which is just another form of accuracy. The problem is just, that it" s more complicated to evaluate a classification model than a regression model. We will talk about this in the following section.

## Further Evaluation

Confusion Matrix:

```
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
predictions = cross_val_predict(random_forest, X_train, Y_train,
cv=3) confusion_matrix(Y_train, predictions)
```

```
array([[488,  61],
       [ 95, 247]])
```

The first row is about the not-survived-predictions: **493 passengers were correctly classified as not survived** (called true negatives) and **56 where wrongly classified as not survived** (false positives).

The second row is about the survived-predictions: **93 passengers where wrongly classified as survived** (false negatives) and **249 where correctly classified as survived** (true positives).

A confusion matrix gives you a lot of information about how well your model does, but theres a way to get even more, like computing the classifiers precision.

Precision and Recall:

```
from sklearn.metrics import precision_score, recall_score

print("Precision:", precision_score(Y_train, predictions))
print("Recall:",recall_score(Y_train, predictions))
```

Precision: 0.801948051948
Recall: 0.722222222222

Our model predicts 81% of the time, a passengers survival correctly (precision). The recall tells us that it predicted the survival of 73 % of the people who actually survived.

## F-Score

You can combine precision and recall into one score, which is called the F- score. The F-score is computed with the harmonic mean of precision and recall. Note that it assigns much more weight to low values. As a result of that, the classifier will only get a high F-score, if both recall and precision are high.

```
from sklearn.metrics import
f1_score f1_score(Y_train,
```

0.7599999999999

There we have it, a 77 % F-score. The score is not that high, because we have a recall of 73%. But unfortunately the F-score is not perfect, because it favors classifiers that have a similar precision and recall. This is a problem, because you sometimes want a high precision and sometimes a high recall. The thing is that an increasing precision, sometimes results in an decreasing recall and vice versa (depending on the threshold). This is called the precision/recall tradeoff. We will discuss this in the following section.

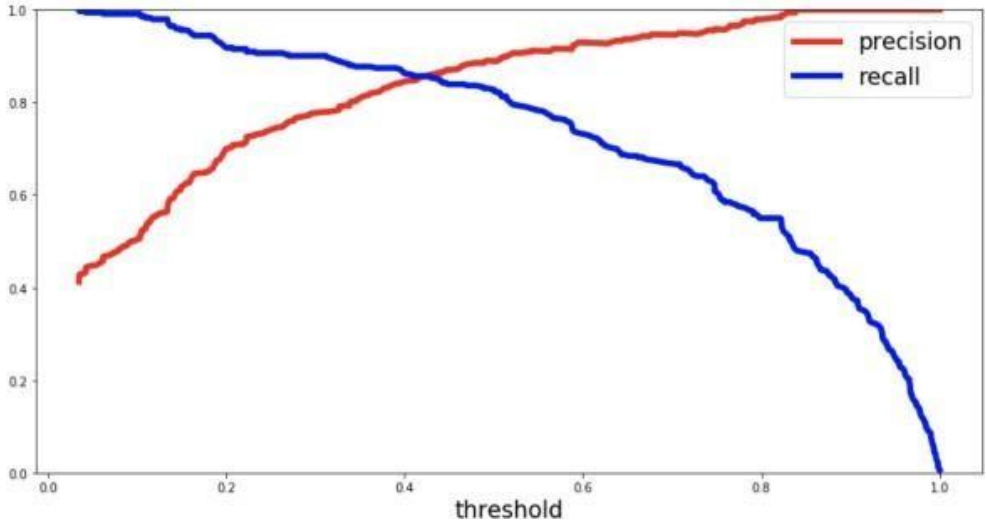
## Precision Recall Curve

For each person the Random Forest algorithm has to classify, it computes a probability based on a function and it classifies the person as survived (when the score is bigger the than threshold) or as not survived (when the score is smaller than the threshold). That" s why the threshold plays an important part.

We will plot the precision and recall with the threshold using matplotlib:

```
from sklearn.metrics import precision_recall_curve

# getting the probabilities of our predictions
y_scores = random_forest.predict_proba(X_train)
y_scores = y_scores[:,1]

precision, recall, threshold = precision recall curve(Y train,
```

```
y_scores)def plot_precision_and_recall(precision, recall,
    threshold): plt.plot(threshold, precision[:-1], "r-",
    label="precision",
linewidth=5)
    plt.plot(threshold, recall[:-1], "b", label="recall",
    linewidth=5) plt.xlabel("threshold", fontsize=19)
    plt.legend(loc="upper right", fontsize=19)
    plt.ylim([0, 1])

plt.figure(figsize=(14, 7))
plot_precision_and_recall(precision, recall, threshold)
```

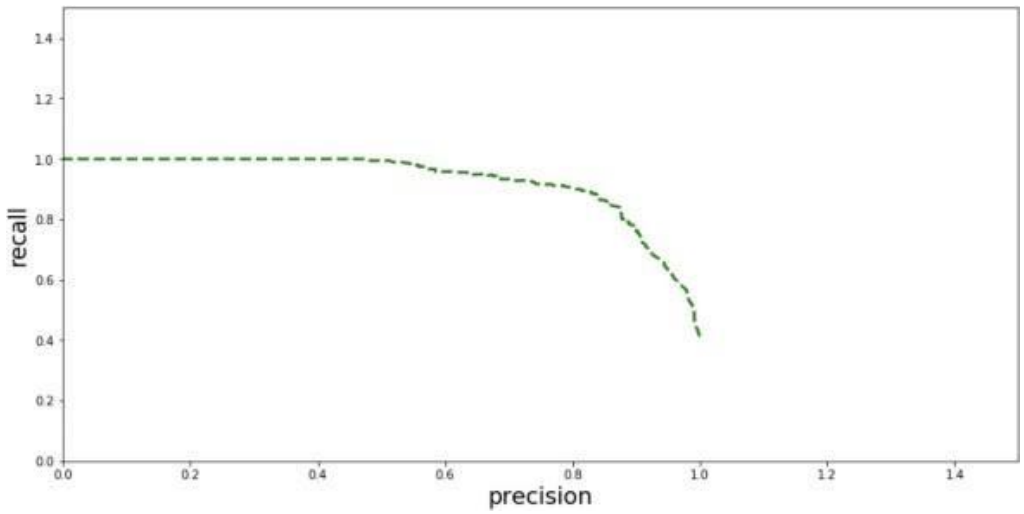



Above you can clearly see that the recall is falling of rapidly at a precision of around 85%. Because of that you may want to select the precision/recall tradeoff before that — maybe at around 75 %.

You are now able to choose a threshold, that gives you the best precision/recall tradeoff for your current machine learning problem. If you want for example a precision of 80%, you can easily look at the plots and see that you would need a threshold of around 0.4. Then you could train a model with exactly that threshold and would get the desired accuracy.

Another way is to plot the precision and recall against each other:

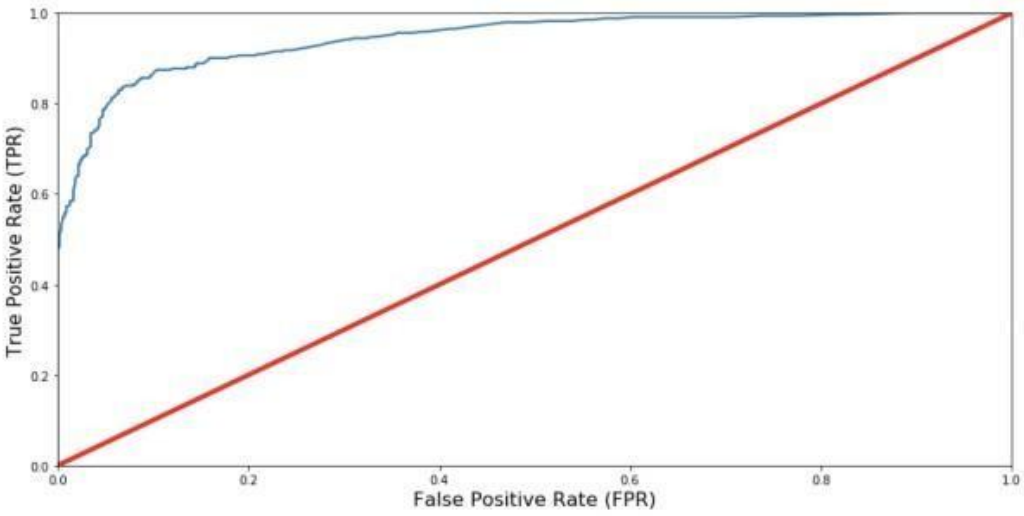
```
def plot_precision_vs_recall(precision, recall):
    plt.plot(recall, precision, "g--", linewidth=2.5)
    plt.ylabel("recall", fontsize=19)
    plt.xlabel("precision", fontsize=19)
    plt.axis([0, 1.5, 0, 1.5])
```




## ROC AUC Curve

Another way to evaluate and compare your binary classifier is provided by the ROC AUC Curve. This curve plots the true positive rate (also called recall) against the false positive rate (ratio of incorrectly classified negative instances), instead of plotting the precision versus

```
from sklearn.metrics import roc_curve
# compute true positive rate and false positive rate
false_positive_rate, true_positive_rate, thresholds =
roc_curve(Y_train, y_scores)# plotting them against each
other def plot_roc_curve(false_positive_rate,
true_positive_rate, label=None):
    plt.plot(false_positive_rate, true_positive_rate, linewidth=2,
label=label)
    plt.plot([0, 1], [0, 1], 'r', linewidth=4)
    plt.axis([0, 1, 0, 1])
    plt.xlabel('False Positive Rate (FPR)',
    fontsize=16) plt.ylabel('True Positive Rate
    (TPR)', fontsize=16)

plt.figure(figsize=(14, 7))
plot_roc_curve(false_positive_rate, true_positive_rate)
```



The red line in the middel represents a purely random classifier (e.g a coin flip) and therefore your classifier should be as far away from it as possible. Our Random Forest model seems to do a good job.

Of course we also have a tradeoff here, because the classifier produces more false positives, the higher the true positive rate is.

## ROC AUC Score

The ROC AUC Score is the corresponding score to the ROC AUC Curve. It is simply computed by measuring the area under the curve, which is called AUC.

A classifiers that is 100% correct, would have a ROC AUC Score of 1 and a completely random classiffier would have a score of 0.5.

```
from sklearn.metrics import
roc_auc_score r_a_score =
roc_auc_score(Y_train, y_scores)
```

ROC_AUC_SCORE: 0.945067587

# Group C
# Assignment No : 13

**Title of the Assignment:** Installation of MetaMask and study spending Ether per transaction

**Objective of the Assignment:** Students should be able to learn new technology such as metamask.Its application and implementations

**Prerequisite:**

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain

---

**Contents for Theory:**

1. **Introduction Blockchain**
2. **Cryptocurrency**
3. **Transaction Wallets**
4. **Ether transaction**
5. **Installation Process of Metamask**

---

**Introduction to Blockchain**

- Blockchain can be described as a data structure that holds transactional records and while ensuring security, transparency, and decentralization. You can also think of it as a chain or records stored in the forms of blocks which are controlled by no single authority.
- A blockchain is a distributed ledger that is completely open to any and everyone on the network. Once an information is stored on a blockchain, it is extremely difficult to change or alter it.
- Each transaction on a blockchain is secured with a digital signature that proves its authenticity. Due to the use of encryption and digital signatures, the data stored on the blockchain is tamper-proof and cannot be changed.
- Blockchain technology allows all the network participants to reach an agreement, commonly known as consensus. All the data stored on a blockchain is recorded digitally and has a common history which is available for all the network participants. This way, the chances of any fraudulent activity or duplication of transactions is eliminated without the need of a third-party.

**Blockchain Features**

The following features make the revolutionary technology of blockchain stand out:

- **Decentralized**

  Blockchains are decentralized in nature meaning that no single person or group holds the authority of the overall network. While everybody in the network has the copy of the distributed ledger with them, no one can modify it on his or her own. This unique feature of blockchain allows transparency and security while giving power to the users.

- **Peer-to-Peer Network**

  With the use of Blockchain, the interaction between two parties through a peer-to-peer model is easily accomplished without the requirement of any third party. Blockchain uses P2P protocol which allows all the network participants to hold an identical copy of transactions, enabling approval through a machine consensus. For example, if you wish to make any transaction from one part of the world to another, you can do that with blockchain all by yourself within a few seconds. Moreover, any interruptions or extra charges will not be deducted in the transfer.

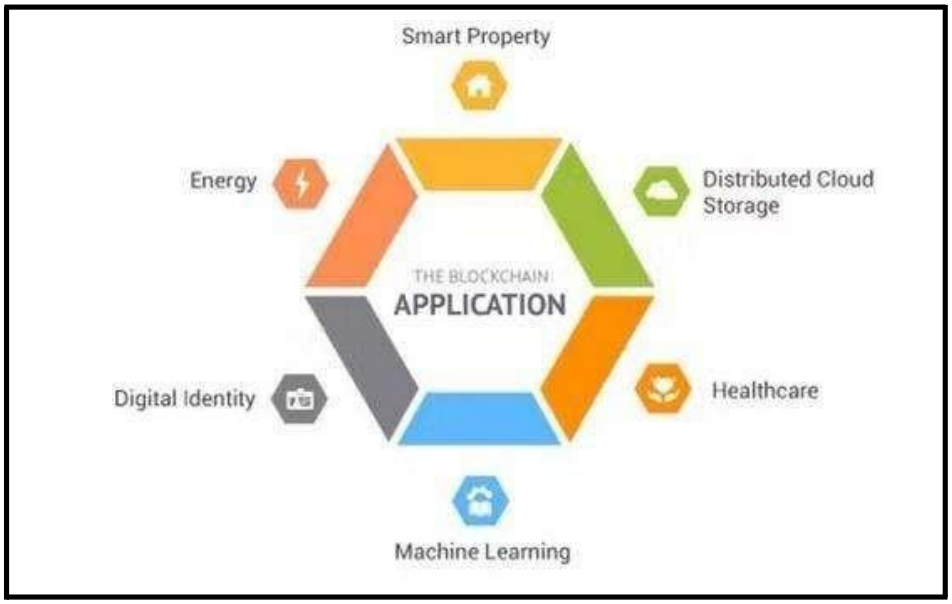
- **Immutable**

  The immutability property of a blockchain refers to the fact that any data once written on the blockchain cannot be changed. To understand immutability, consider sending email as an example. Once you send an email to a bunch of people, you cannot take it back. In order to find a way around, you'll have to ask all the recipients to delete your email which is pretty tedious. This is how immutability works.

- **Tamper-Proof**

  With the property of immutability embedded in blockchains, it becomes easier to detect tampering of any data. Blockchains are considered tamper-proof as any change in even one single block can be detected and addressed smoothly. There are two key ways of detecting tampering namely, hashes and blocks.

**Popular Applications of Blockchain Technology**




**Benefits of Blockchain Technology:**

- **Time-saving:** No central Authority verification needed for settlements making the process faster and cheaper.
- **Cost-saving:** A Blockchain network reduces expenses in several ways. No need for third-party verification. Participants can share assets directly. Intermediaries are reduced. Transaction efforts are minimized as every participant has a copy of shared ledger.
- **Tighter security:** No one can temper with Blockchain Data as it is shared among

millions of participants. The system is safe against cybercrimes and Fraud.

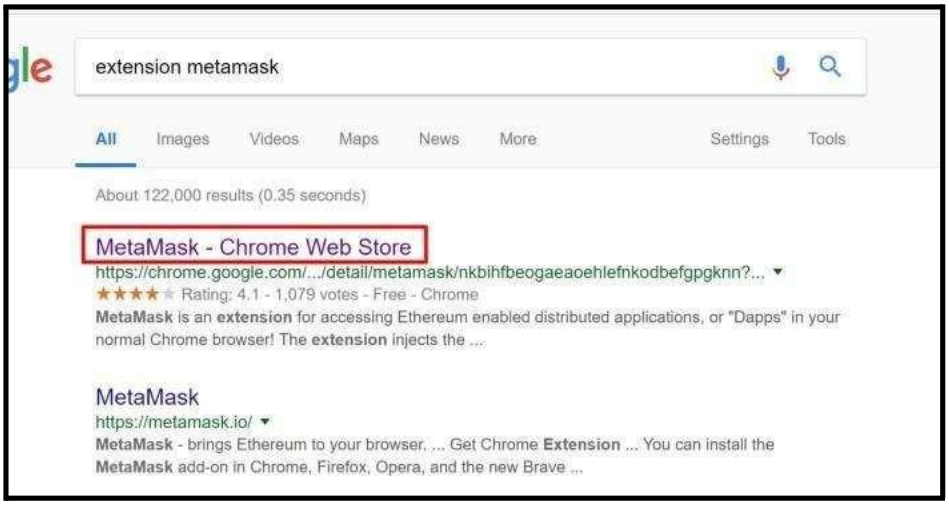
- In finance market trading, Fibonacci retracement levels are widely used in technical analysis.

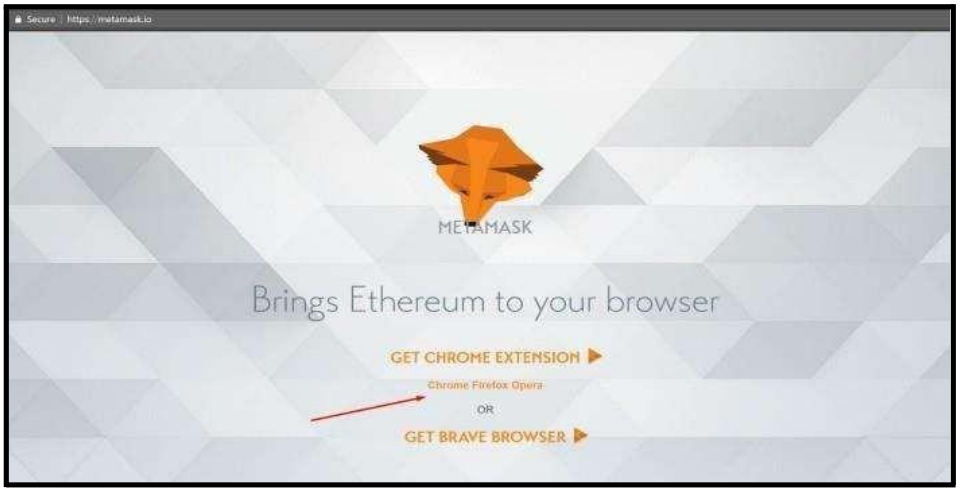
### How to use MetaMask: A step by step guide

MetaMask is one of the most popular browser extensions that serves as a way of storing your Ethereum and other ERC-20 Tokens. The extension is free and secure, allowing web applications to read and interact with Ethereum's blockchain.

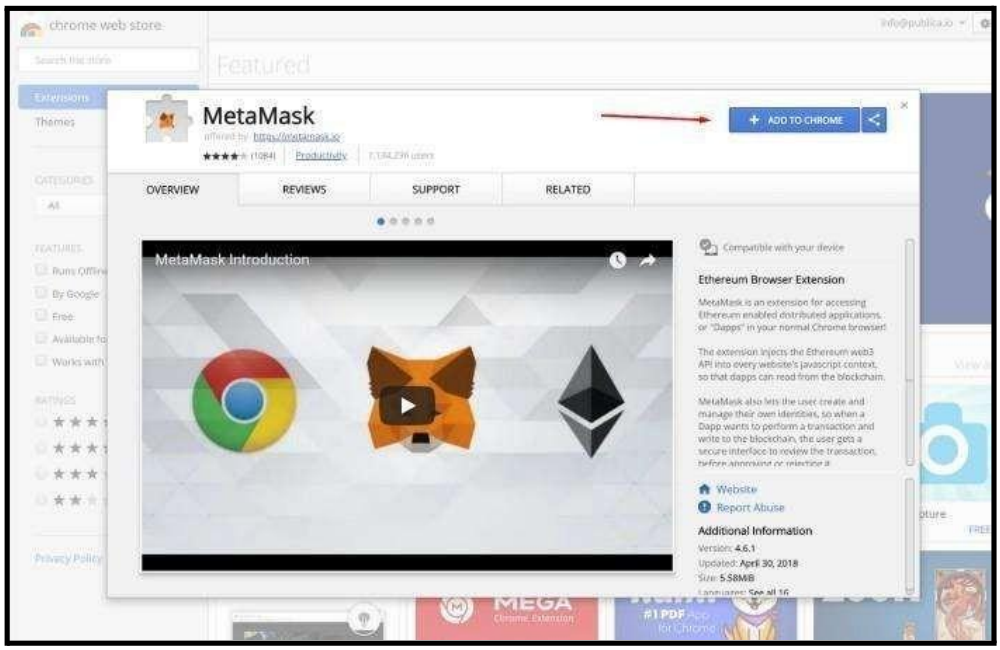
## Step 1. Install MetaMask on your browser.

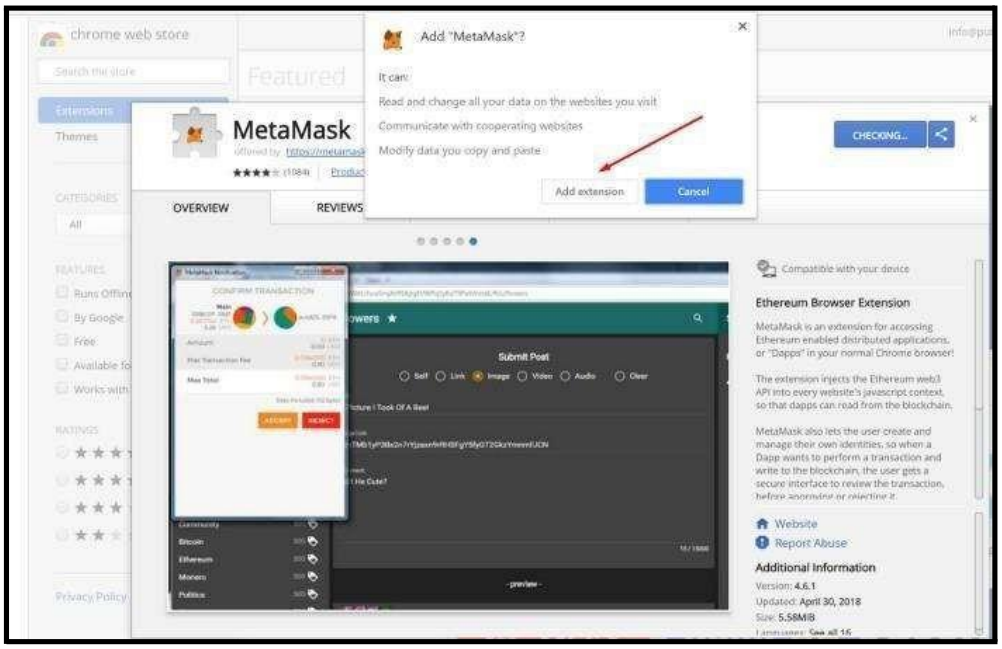
To create a new wallet, you have to install the extension first. Depending on your browser, there are different marketplaces to find it. Most browsers have MetaMask on their stores, so it's not that hard to see it, but either way, here they are Chrome, Firefox, and Opera.






- Click on **Install MetaMask** as a Google Chrome extension.
- Click **Add to Chrome**.
- Click **Add Extension**.

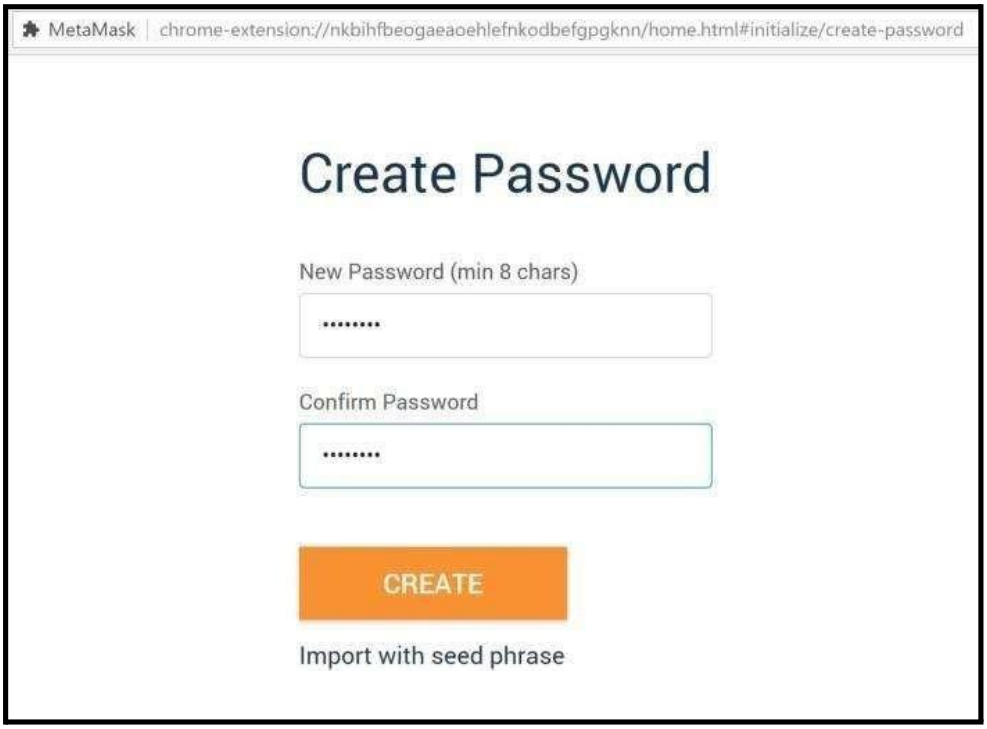





nd it's as easy as that to install the extension on your browser, continue reading the next step to figure out how to create an account.

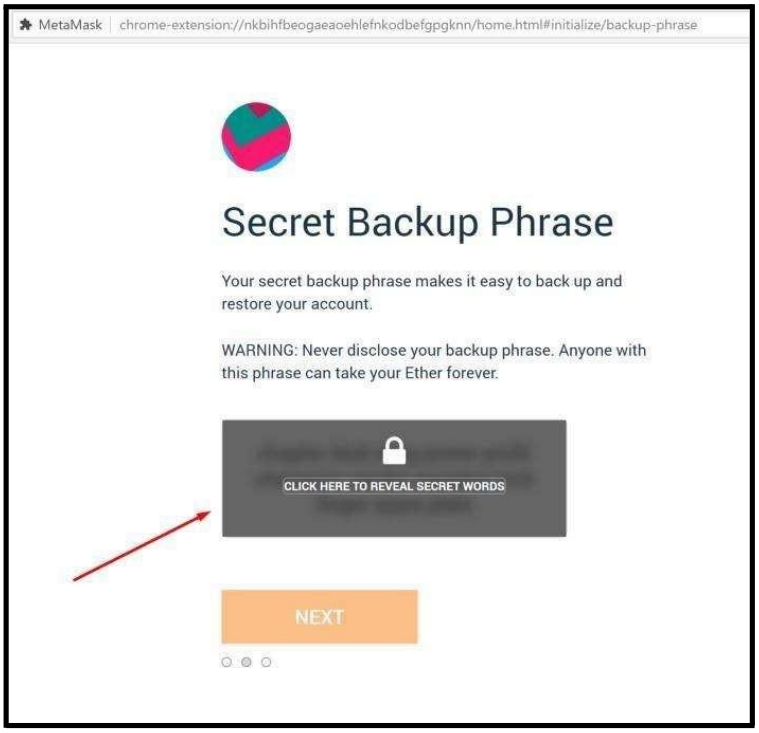
## Step 2. Create an account.

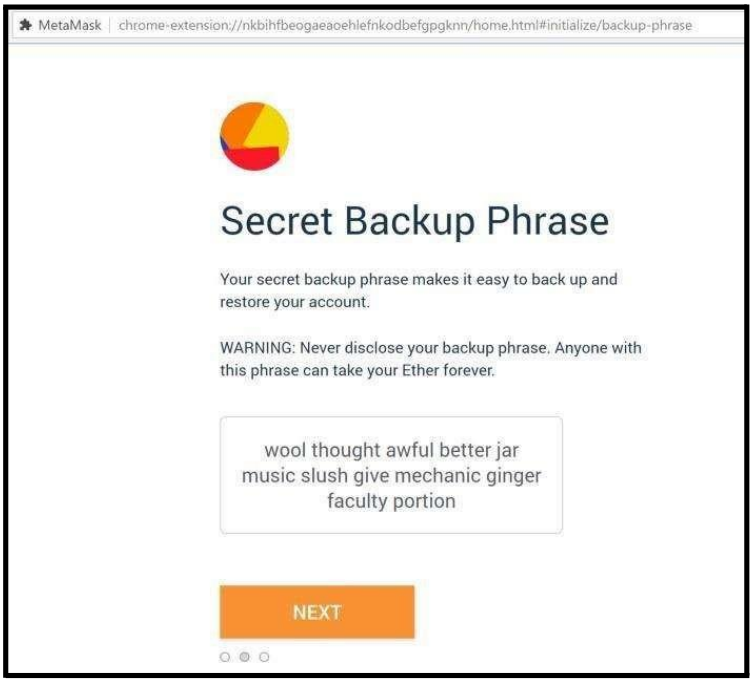
- Click on the extension icon in the upper right corner to open MetaMask.
- To install the latest version and be up to date, **click Try it now**.
- **Click Continue**.
- You will be prompted to create a new password. **Click Create**.




- Proceed by **clicking Next** and accept the Terms of Use.

**Click Reveal Secret Words**. There you will see a 12 words seed phrase. This is really important and usually not a good idea to store digitally, so take your time and write it down



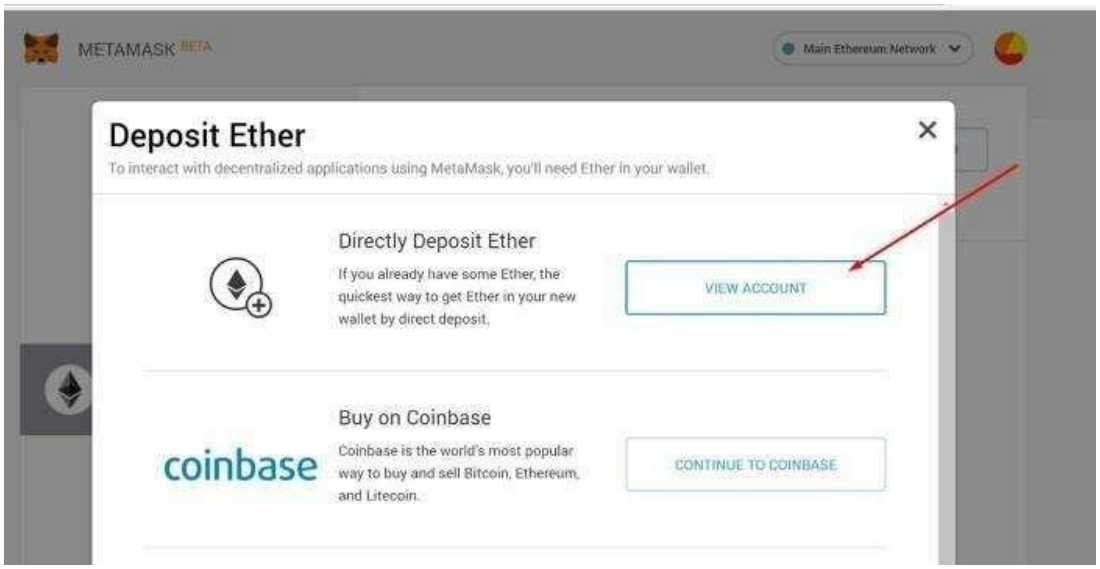

- Verify your secret phrase by selecting the previously generated phrase in order. **Click Confirm**.

And that's it; now you have created your MetaMask account successfully. A new Ethereum wallet

address has just been created for you. It's waiting for you to deposit funds, and if you want to learn how to do that, look at the next step below.

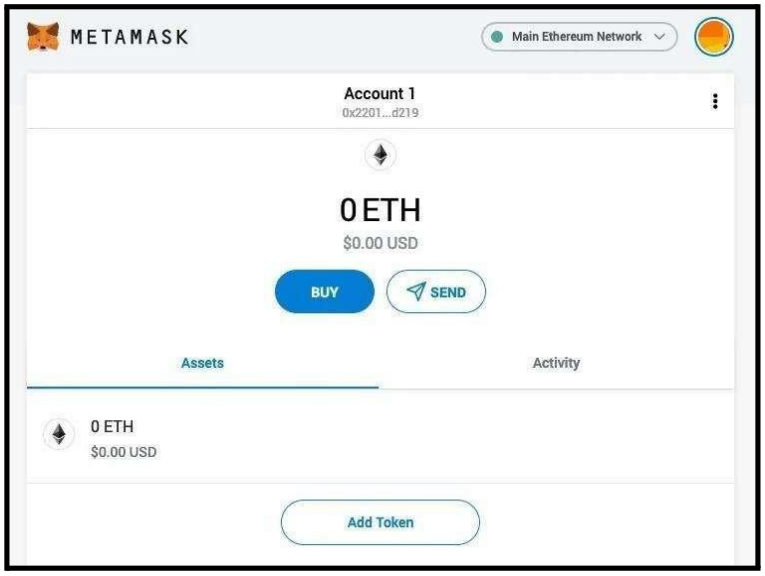
## Step 3. Depositing funds.

- Click on **View Account**.



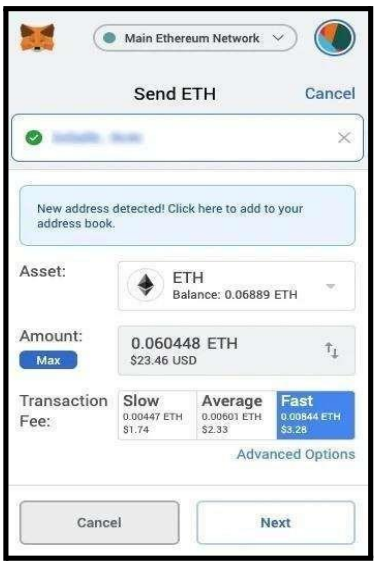

You can now see your public address and share it with other people. There are some methods to buy coins offered by MetaMask, but you can do it differently as well; you just need your address.

If you ever get logged out, you'll be able to log back in again by clicking the MetaMask icon, which will have been added to your web browser (usually found next to the URL bar).




You can now access your list of assets in the 'Assets' tab and view your transaction history in the 'Activity' tab.

- Sending crypto is as simple as clicking the 'Send' button, entering the recipient address and amount to send, and selecting a transaction fee. You can also manually adjust the transaction fee using the 'Advanced Options' button, using information from ETH Gas Station or similar platforms to choose a more acceptable gas price.
- After clicking 'Next', you will then be able to either confirm or reject the transaction on the subsequent page.



- To use MetaMask to interact with a dapp or smart contract, you'll usually need to find a 'Connect to Wallet' button or similar element on the platform you are trying to use. After clicking this, you should then see a prompt asking whether you want to let the dapp connect to your wallet.

## What advantages does MetaMask have?

- **Popular** - It is commonly used, so users only need one plugin to access a wide range of dapps.
- **Simple** - Instead of managing private keys, users just need to remember a list of words, and transactions are signed on their behalf.
- **Saves space** - Users don't have to download the Ethereum blockchain, as MetaMask sends requests to nodes outside of the user's computer.
- **Integrated** - Dapps are designed to work with MetaMask, so it becomes much easier to send Ether in and out.

**Conclusion-** In this way we have explored Concept Blockchain and metamat wallet for transaction of digital currency

**Assignment Question**

1. **What Are the Different Types of Blockchain Technology?**
2. **What Are the Key Features/Properties of Blockchain?**
3. **What Type of Records You Can Keep in A Blockchain?**
4. **What is the difference between Ethereum and Bitcoin?**
5. **What are Merkle Trees? Explain their concept.**
6. **What is Double Spending in transaction operation**
7. **Give real-life use cases of blockchain.**

**Reference link**


- https://hackernoon.com/blockchain-technology-explained-introduction-meaning-and-applications-edbd6759a2b2
- https://levelup.gitconnected.com/how-to-use-metamask-a-step-by-step-guide-f380a3943fb1
- https://decrypt.co/resources/metamask

# Assignment No : 14

**Title of the Assignment:** Create your own wallet using Metamask for crypto transactions

**Objective of the Assignment:** Students should be able to learn about cryptocurrencies and learn how transaction done by using different digital currency

**Prerequisite:**

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain

---

**Contents for Theory:**

1. **Cryptocurrency**
2. **Transaction Wallets**
3. **Ether transaction**

---

**Introduction to Cryptocurrency**

- Cryptocurrency is a digital payment system that doesn't rely on banks to verify transactions. It's a peer-to-peer system that can enable anyone anywhere to send and receive payments. Instead of being physical money carried around and exchanged in the real world, cryptocurrency payments exist purely as digital entries to an online database describing specific transactions. When you transfer cryptocurrency funds, the transactions are recorded in a public ledger. Cryptocurrency is stored in digital wallets.
- Cryptocurrency received its name because it uses encryption to verify transactions. This means advanced coding is involved in storing and transmitting cryptocurrency data between wallets and to public ledgers. The aim of encryption is to provide security and safety.
- The first cryptocurrency was Bitcoin, which was founded in 2009 and remains the best known today. Much of the interest in cryptocurrencies is to trade for profit, with speculators at times driving prices skyward.

**How does cryptocurrency work?**

- Cryptocurrencies run on a distributed public ledger called blockchain, a record of all transactions updated and held by currency holders.
- Units of cryptocurrency are created through a process called mining, which involves using computer power to solve complicated mathematical problems that generate coins. Users can also buy the currencies from brokers, then store and spend them using cryptographic wallets.
- If you own cryptocurrency, you don't own anything tangible. What you own is a key that allows you to move a record or a unit of measure from one person to another without a trusted third party.
- Although Bitcoin has been around since 2009, cryptocurrencies and applications of blockchain technology are still emerging in financial terms, and more uses are expected in the future. Transactions including bonds, stocks, and other financial assets could eventually be traded using the technology.

**Cryptocurrency examples**

There are thousands of cryptocurrencies. Some of the best known include:

- **Bitcoin:**

Founded in 2009, Bitcoin was the first cryptocurrency and is still the most commonly traded. The currency was developed by Satoshi Nakamoto – widely believed to be a pseudonym for an individual or group of people whose precise identity remains unknown.

- **Ethereum:**

  Developed in 2015, Ethereum is a blockchain platform with its own cryptocurrency, called Ether (ETH) or Ethereum. It is the most popular cryptocurrency after Bitcoin.

- **Litecoin:**

  This currency is most similar to bitcoin but has moved more quickly to develop new innovations, including faster payments and processes to allow more transactions.

- **Ripple:**

  Ripple is a distributed ledger system that was founded in 2012. Ripple can be used to track different kinds of transactions, not just cryptocurrency. The company behind it has worked with various banks and financial institutions.

- Non-Bitcoin cryptocurrencies are collectively known as “altcoins” to distinguish them from the original.

**How to store cryptocurrency**

- Once you have purchased cryptocurrency, you need to store it safely to protect it from hacks or theft. Usually, cryptocurrency is stored in crypto wallets, which are physical devices or online software used to store the private keys to your cryptocurrencies securely. Some exchanges provide wallet services, making it easy for you to store directly through the platform. However, not all exchanges or brokers automatically provide wallet services for you.
- There are different wallet providers to choose from. The terms “hot wallet” and “cold wallet” are used:
- **Hot wallet storage:** "hot wallets" refer to crypto storage that uses online software to protect the private keys to your assets.
- **Cold wallet storage:** Unlike hot wallets, cold wallets (also known as hardware wallets) rely on offline electronic devices to securely store your private keys.

**Conclusion-** In this way we have explored Concept Cryptocurrency and learn how transactions are done using digital currency

**Assignment Question**

1. **What is Bitcoin?**
2. **What Are the biggest Four common cryptocurrency scams**
3. **Explain How safe are money e-transfers?**
4. **What is cryptojacking and how does it work?**

**Reference link**

- https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency

# Assignment No : 15

**Title of the Assignment:** Write a smart contract on a test network, for Bank account of a customer for following operations:

- Deposit money
- Withdraw Money
- Show balance

**Objective of the Assignment:** Students should be able to learn new technology such as metamask.Its application and implementations

**Prerequisite:**

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain.

**Contents for Theory:**

The contract will allow deposits from any account, and can be trusted to allow withdrawals only by accounts that have sufficient funds to cover the requested withdrawal.

This post assumes that you are comfortable with the ether-handling concepts introduced in our post, Writing a Contract That Handles Ether.

That post demonstrated how to restrict ether withdrawals to an "owner's" account. It did this by persistently storing the owner account's address, and then comparing it to the msg.sender value for any withdrawal attempt. Here's a slightly simplified version of that smart contract, which allows anybody to deposit money, but only allows the owner to make withdrawals:

pragma solidity ^0.4.19;

```
contract TipJar {

    address owner;    // current owner of the contract

    function TipJar() public {
        owner = msg.sender;
    }

    function withdraw() public {
        require(owner == msg.sender);
        msg.sender.transfer(address(this).balance);
    }

    function deposit(uint256 amount) public payable {
        require(msg.value == amount);
    }

    function getBalance() public view returns (uint256) {
        return address(this).balance;
    }
}
```

I am going to generalize this contract to keep track of ether deposits based on the account address of the depositor, and then only allow that same account to make withdrawals of that ether. To do this, we need a way keep track of account balances for each depositing account—a mapping from accounts to balances. Fortunately, Solidity provides a ready-made mapping data type that can map account addresses to integers,

which will make this bookkeeping job quite simple. (This mapping structure is much more general than just addresses to integers, but that's all we need here.)

Here's the code to accept deposits and track account balances:

```
pragma solidity ^0.4.19;

contract Bank {

    mapping(address => uint256) public balanceOf;  // balances, indexed by addresses

    function deposit(uint256 amount) public payable {
        require(msg.value == amount);

        balanceOf[msg.sender] += amount;     // adjust the account's balance
    }
}
```

Here are the new concepts in the code above:

- mapping(address => uint256) public balanceOf; declares a persistent public variable, balanceOf, that is a mapping from account addresses to 256-bit unsigned integers. Those integers will represent the current balance of ether stored by the contract on behalf of the corresponding address.
- Mappings can be indexed just like arrays/lists/dictionaries/tables in most modern programming languages.
- The value of a missing mapping value is 0. Therefore, we can trust that the beginning balance for all account addresses will effectively be zero prior to the first deposit.

It's important to note that balanceOf keeps track of the ether balances assigned to each account, but it does not actually move any ether anywhere. The bank contract's ether balance is the sum of all the balances of all accounts–only balanceOf tracks how much of that is assigned to each account.

Note also that this contract doesn't need a constructor. There is no persistent state to initialize other than the balanceOf mapping, which already provides default values of 0.

Given the balanceOf mapping from account addresses to ether amounts, the remaining code for a fully-functional bank contract is pretty small. I'll simply add a withdrawal function:

**bank.sol**

```
pragma solidity ^0.4.19;

contract Bank {

    mapping(address => uint256) public balanceOf;  // balances, indexed by addresses

    function deposit(uint256 amount) public payable {
        require(msg.value == amount);
        balanceOf[msg.sender] += amount;     // adjust the account's balance
    }

    function withdraw(uint256 amount) public {
        require(amount <= balanceOf[msg.sender]);
        balanceOf[msg.sender] -= amount;
        msg.sender.transfer(amount);
    }
}
```

The code above demonstrates the following:

- The require(amount <= balances[msg.sender]) checks to make sure the sender has sufficient funds to cover the requested withdrawal. If not, then the transaction aborts without making any state changes or ether transfers.
- The balanceOf mapping must be updated to reflect the lowered residual amount after the withdrawal.
- The funds must be sent to the sender requesting the withdrawal.

In the withdraw() function above, it is very important to adjust balanceOf[msg.sender] **before** transferring ether to avoid an exploitable vulnerability. The reason is specific to smart contracts and the fact that a transfer to a smart contract executes code in that smart contract. (The essentials of Ethereum transactions are discussed in How Ethereum Transactions Work.)

Now, suppose that the code in withdraw() did not adjust balanceOf[msg.sender] before making the transfer *and* suppose that msg.sender was a malicious smart contract. Upon receiving the transfer–handled by msg.sender's fallback function–that malicious contract could initiate *another* withdrawal from the banking contract. When the banking contract handles this second withdrawal request, it would have already transferred ether for the original withdrawal, but it would not have an updated balance, so it would allow this second withdrawal!

This vulnerability is called a "reentrancy" bug because it happens when a smart contract invokes code in a different smart contract that then calls back into the original, thereby reentering the exploitable contract. For this reason, it's essential to always make sure a contract's internal state is fully updated before it potentially invokes code in another smart contract. (And, it's essential to remember that every transfer to a smart contract executes that contract's code.)

To avoid this sort of reentrancy bug, follow the "Checks-Effects-Interactions pattern" as described in the Solidity documentation. The `withdraw()` function above is an example of implementing this pattern

# Assignment No : 16

**Title of the Assignment:** Write a survey report on types of Blockchains and its real time use cases.

**Objective of the Assignment:** Students should be able to learn new technology such as metamask.Its application and implementations

**Prerequisite:**

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain

---

**Contents for Theory:**

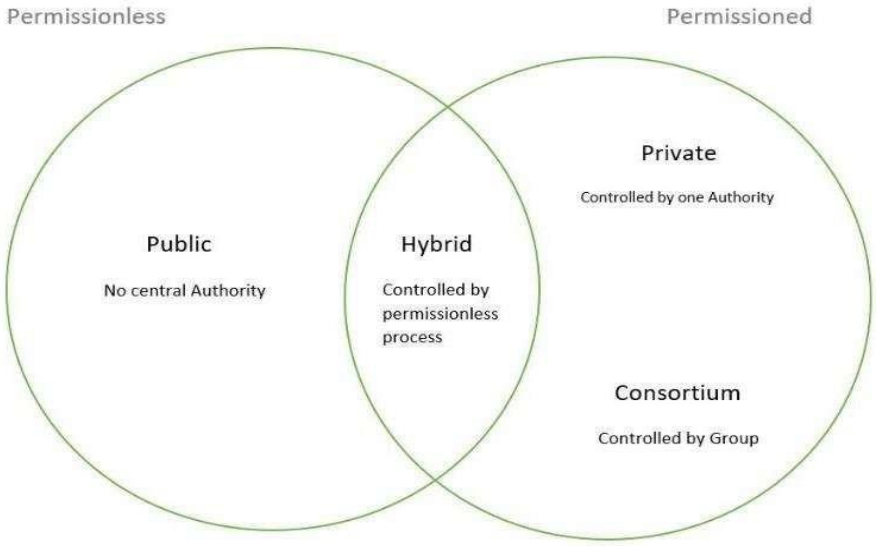
**There are 4 types of blockchain:**

**Public Blockchain.**

**Private Blockchain.**

**Hybrid Blockchain.**

**Consortium Blockchain**




---

1. Public Blockchain



These blockchains are completely open to following the idea of decentralization. They don't have any restrictions, anyone having a computer and internet can participate in the network.

As the name is public this blockchain is open to the public, which means it is not owned by anyone.
Anyone having internet and a computer with good hardware can participate in this public blockchain.
All the computer in the network hold the copy of other nodes or block present in the network
In this public blockchain, we can also perform verification of transactions or records
Advantages:
Trustable: There are algorithms to detect no fraud. Participants need not worry about the other nodes in the network
Secure: This blockchain is large in size as it is open to the public. In a large size, there is greater distribution of records
Anonymous Nature: It is a secure platform to make your transaction properly at the same time, you are not required to reveal your name and identity in order to participate.
Decentralized: There is no single platform that maintains the network, instead every user has a copy of the ledger.
Disadvantages:

Processing: The rate of the transaction process is very slow, due to its large size. Verification of each node is a very time-consuming process.
Energy Consumption: Proof of work is high energy-consuming. It requires good computer hardware to participate in the network
Acceptance: No central authority is there so governments are facing the issue to implement the technology faster.
Use Cases: Public Blockchain is secured with proof of work or proof of stake they can be used to displace traditional financial systems. The more advanced side of this blockchain is the smart contract that enabled this blockchain to support decentralization. Examples of public blockchain are Bitcoin, Ethereum.

2. Private Blockchain
These blockchains are not as decentralized as the public blockchain only selected nodes can participate in the process, making it more secure than the others.

These are not as open as a public blockchain.
They are open to some authorized users only.
These blockchains are operated in a closed network.
In this few people are allowed to participate in a network within a company/organization.
Advantages:

Speed: The rate of the transaction is high, due to its small size. Verification of each node is less time-consuming.
Scalability: We can modify the scalability. The size of the network can be decided manually.
Privacy: It has increased the level of privacy for confidentiality reasons as the businesses required.
Balanced: It is more balanced as only some user has the access to the transaction which improves the performance of the network.
Disadvantages:

Security- The number of nodes in this type is limited so chances of manipulation are there. These blockchains are more vulnerable.
Centralized- Trust building is one of the main disadvantages due to its central nature. Organizations can use this for malpractices.
Count- Since there are few nodes if nodes go offline the entire system of blockchain can be endangered.
Use Cases: With proper security and maintenance, this blockchain is a great asset to secure information without exposing it to the public eye. Therefore companies use them for internal auditing, voting, and asset management. An example of private blockchains is Hyperledger, Corda.

3. Hybrid Blockchain
It is the mixed content of the private and public blockchain, where some part is controlled by some organization and other makes are made visible as a public blockchain.
It is a combination of both public and private blockchain. Permission-based and permissionl ess systems are used. User access information via smart contracts
Even a primary entity owns a hybrid blockchain it cannot alter the transaction

Advantages:
Ecosystem: Most advantageous thing about this blockchain is its hybrid nature. It cannot be hacked as 51% of users don't have access to the network
Cost: Transactions are cheap as only a few nodes verify the transaction. All the nodes don't carry the verification hence less computational cost.
Architecture: It is highly customizable and still maintains integrity, security, and transparency.
Operations: It can choose the participants in the blockchain and decide which transaction can be made public.

Disadvantages:

Efficiency: Not everyone is in the position to implement a hybrid Blockchain. The organization also faces some difficulty in terms of efficiency in maintenance.
Transparency: There is a possibility that someone can hide information from the user. If someone wants to get access through a hybrid blockchain it depends on the organization whether they will give or not.
Ecosystem: Due to its closed ecosystem this blockchain lacks the incentives for network participation.
Use Case: It provides a greater solution to the health care industry, government, real estate, and financial companies. It provides a remedy where data is to be accessed publicly but needs to be shielded privately.
Examples of Hybrid Blockchain are Ripple network and XRP token.

4. Consortium Blockchain
It is a creative approach that solves the needs of the organization. This blockchain validates the transaction and also initiates or receives transactions.

Also known as Federated Blockchain.
This is an innovative method to solve the organization's needs.
Some part is public and some part is private.
In this type, more than one organization manages the blockchain.
Advantages:

Speed: A limited number of users make verification fast. The high speed makes this more usable for organizations.
Authority: Multiple organizations can take part and make it decentralized at every level. Decentralized authority, makes it more secure.
Privacy: The information of the checked blocks is unknown to the public view. but any member belonging to the blockchain can access it.
Flexible: There is much divergence in the flexibility of the blockchain. Since it is not a very large decision can be taken faster.
Disadvantages:

Approval: All the members approve the protocol making it less flexible. Since one or more organizations are involved there can be differences in the vision of interest.
Transparency: It can be hacked if the organization becomes corrupt. Organizations may hide information from the users.
Vulnerability: If few nodes are getting compromised there is a greater chance of vulnerability in this blockchain
Use Cases: It has high potential in businesses, banks, and other payment processors. Food tracking of the organizations frequently collaborates with their sectors making it a federated solution ideal for their use.
Examples of consortium Blockchain are Tendermint and Multichain.

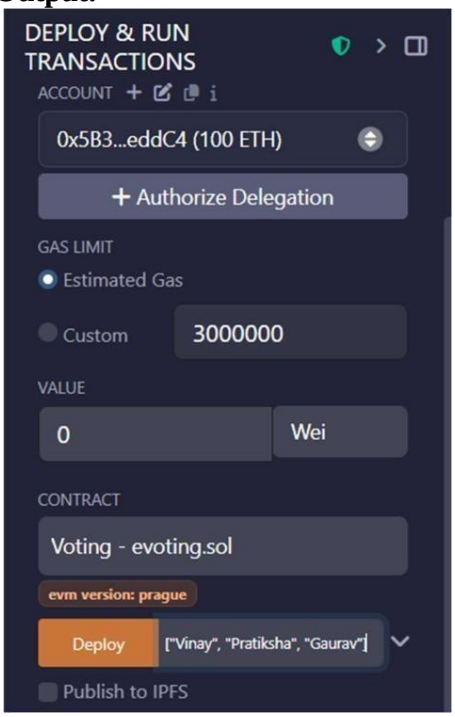
**Conclusion**-In this way we have explored types of blockchain and its applications in real time

**MINI PROJECT**

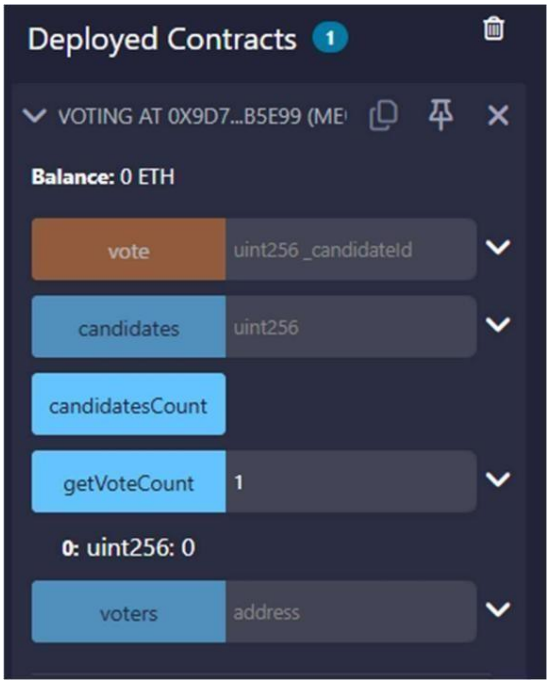
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

/**
* @title E-Voting System
* @dev A simple smart contract for conducting an election.
 */
contract Voting {

    // Structure to represent a candidate     struct
Candidate {
        uint id;
string name;
        uint voteCount;
    }

    // Structure to represent a voter
struct Voter {         bool
isRegistered;          bool hasVoted;
        uint votedFor; // The ID of the candidate they voted for
    }

    // Mapping to store candidates by their ID
 mapping(uint => Candidate) public candidates;

     // Mapping to store voters by their address
     mapping(address => Voter) public voters;

    // Counter to keep track of the total number of candidates      uint
public candidatesCount;

    /**
* @dev Constructor to initialize the election with a list of candidate names.       * The person who
   deploys the contract is the administrator.
     */
    constructor(string[] memory _candidateNames) {
for (uint i = 0; i < _candidateNames.length; i++) {
            addCandidate(_candidateNames[i]);
        }
    }

    /**
* @dev Private function to add a new candidate.
     */
    function addCandidate(string memory _name) private {
candidatesCount++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
```

```
    }

    /**
* @dev Allows a voter to cast their vote.
* Requires the voter has not already voted.
     */
    function vote(uint _candidateId) public {
// Check if the voter has already voted
        require(!voters[msg.sender].hasVoted, "You have already voted.");

        // Check if the candidate ID is valid
        require(_candidateId > 0 && _candidateId <= candidatesCount, "Invalid candidate ID.");

        // Mark the voter as having voted
voters[msg.sender].hasVoted = true;
voters[msg.sender].votedFor = _candidateId;

        // Increment the candidate's vote count
candidates[_candidateId].voteCount++;
    }

    /**
     * @dev Retrieves the total vote count for a specific candidate.
     */
    function getVoteCount(uint _candidateId) public view returns (uint) {
        require(_candidateId > 0 && _candidateId <= candidatesCount, "Invalid candidate ID.");
return candidates[_candidateId].voteCount;
    }
}
```

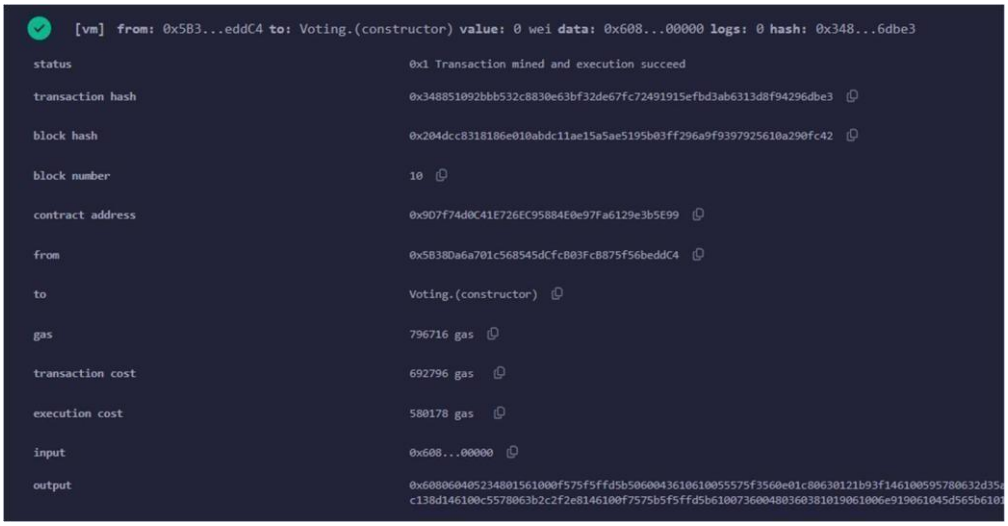
**Output:**



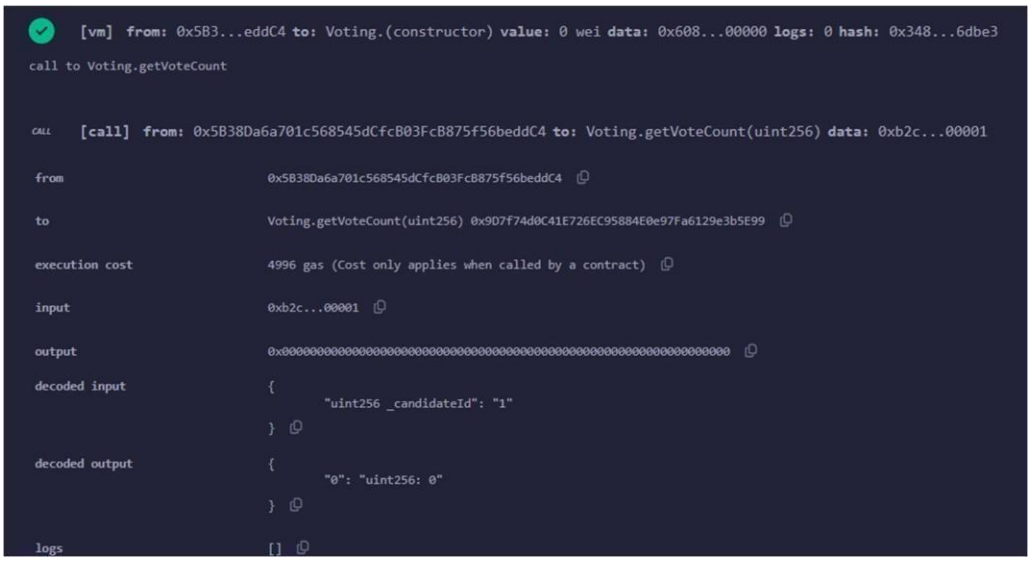

Initial Candidate




Initial Votes

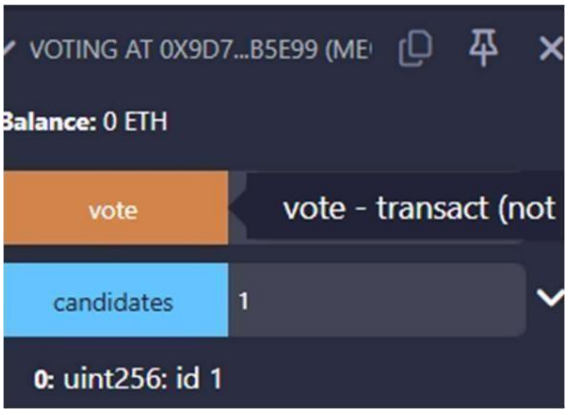



Added Candidates Transaction 1.1

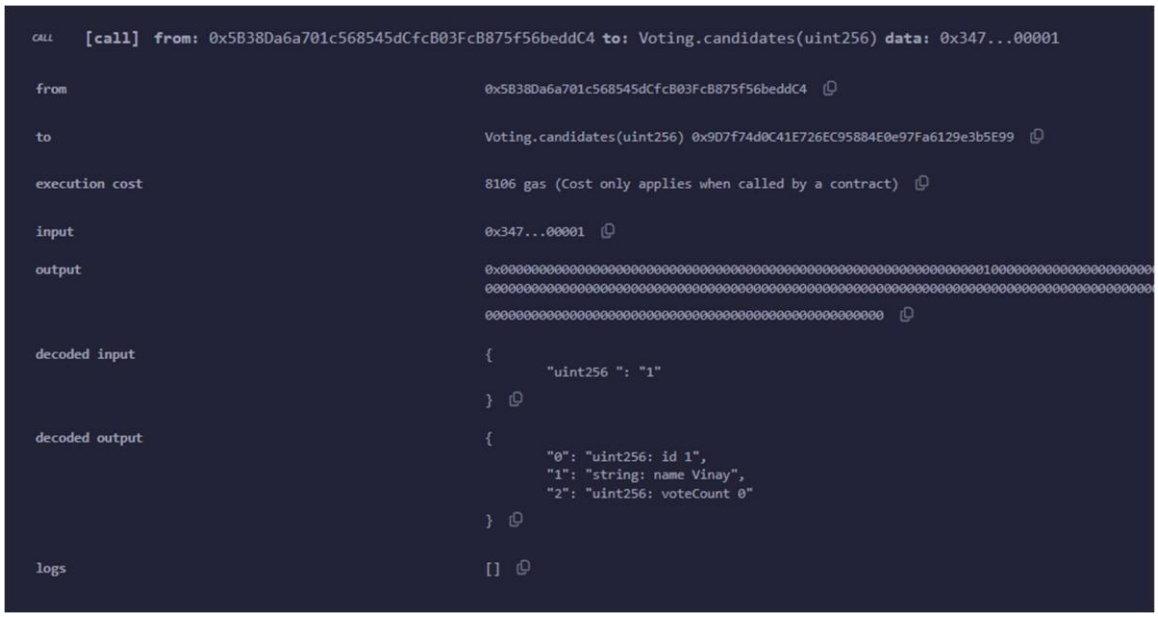



Initial Votes Transaction 1.2

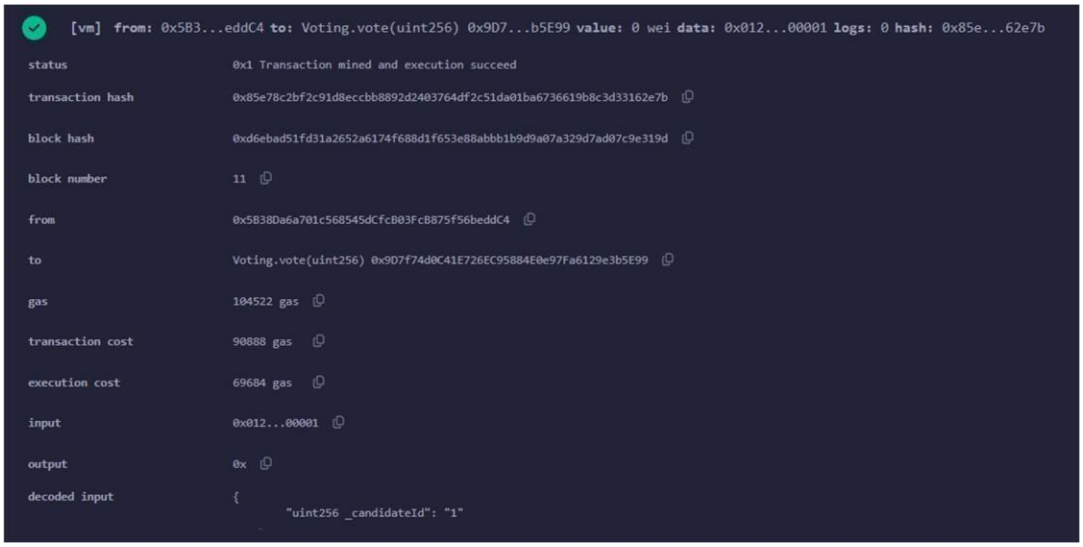

First Vote



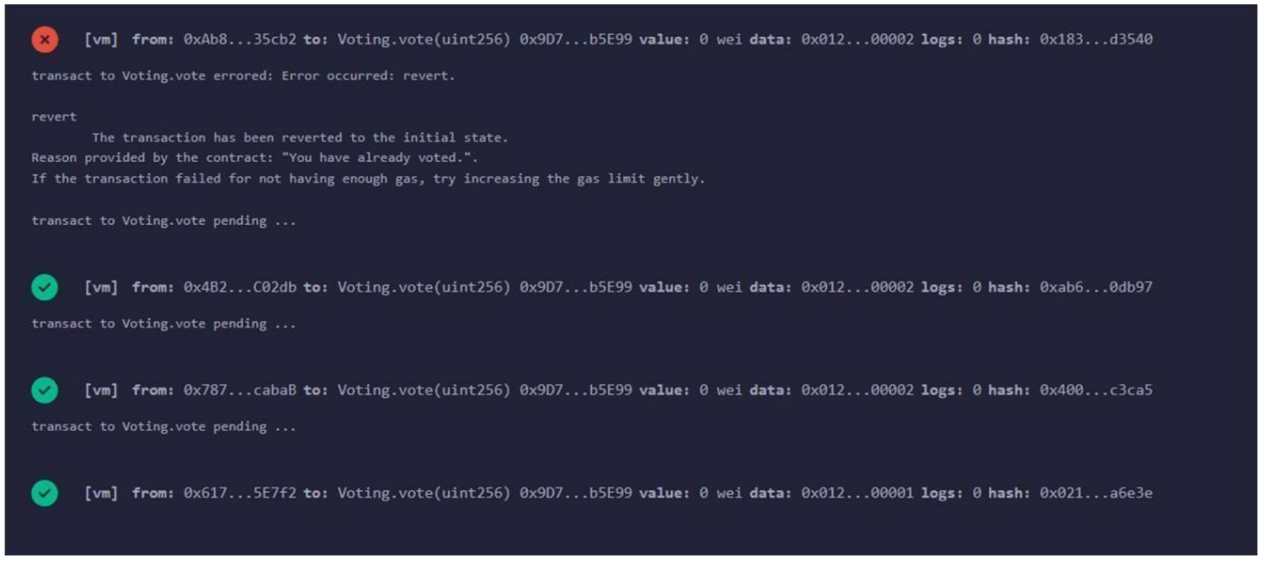

First Vote Count




First Vote Transaction 2.1



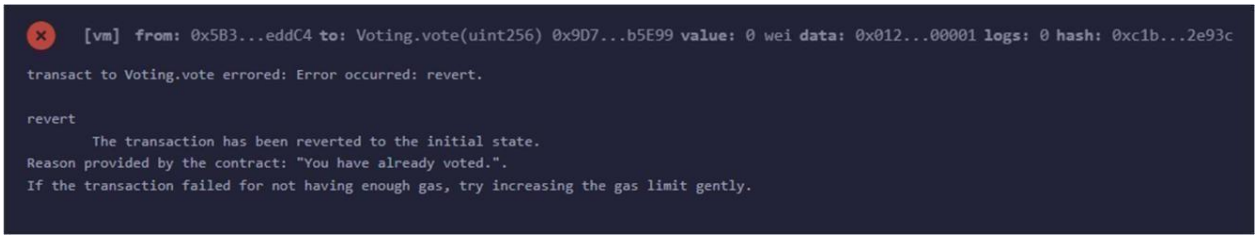

Valid Vote Transaction 2.2




Subsequent Votes Transaction 3.1


Final Vote Count ( Winner ID-1 “Darshan”)


Winner Transaction 3.2




Already Voted Transaction 3.3

## Laboratory Practice - IV

## Elective III - 410244(C): Cyber Security and Digital Forensics

# CSDF

**Prepaid By:**

**Prof.Rajendra Kokare**

**DHOLE PATIL COLLEGE OF ENGINEERING, PUNE**

**DEPARTMENT OF COMPUTER ENGINEERING**

# CERTIFICATE

**This is to certify that student _____is studying in BE Computer Engineering has successfully Submitted and Completed CSDF Lab Manual This study is a partial fulfillment of the degree of Bachelor of Engineering in Computer Engineering of the Savitribai Phule Pune University, Pune during the academic year 2025-2026.**

**Date:**

Place: PRN No:

Date: Exam Seat No:

Subject Teacher Head of Department

**INDEX**

| Sr. No. | Title | Date | |
|---|---|---|---|
| | Any 5 Assignment from Group 1 And 1 Mini project from Group2 is mandatory | | |
| 1 | Write a program for Tracking Emails and Investigating Email Crimes. i.e. Write a program to analyze e–mail header | | |
| 2 | Implement a program to generate and verify CAPTCHA Image. | | |
| 3 | Write a computer forensic application program for Recovering Permanent Deleted Files and Deleted Partitions. | | |
| 4 | Write a program for Log Capturing and Event Correlation | | |
| 5 | Study of Honeypot. | | |
| 6 | To implement a basic function of Code Division Multiple Access (CDMA) to test the orthogonally and autocorrelation of a code to be used for CDMA operation. Write an Application based on the above concept. | | |

**GROUP-2**

| | |
|---|---|
| **1** | Mini–project: Perform the following steps:<br>• Go to the National Child Exploitation Coordination Centre (NCECC) Web site at http:// www.ncecc.ca<br>• Click on the Reporting child exploitation link.<br>• c. Read "How to Report Internet Pornography or Internet Luring Related to Children." |
| **2** | Mini- Project: Perform the following steps:<br>• Go to http://www.usdoj.gov/criminal/cybercrime/cyberstalking.htm.<br>• b. Read the 1999 report on cyber stalking |

# Laboratory Practice IV

# Elective III - 410244(C): Cyber Security and Digital Forensics

# Experiment No: Group A-1

## Problem Definition:

Write a program for Tracking Emails and Investigating Email Crimes. i.e. Write a program to analyze e–mail header

### 1.1 Prerequisite:

Application Layer Protocols

### 1.2 Learning Objective:

1. To understand how Mails are transferred from Sender to Receiver.
2. To Understand Email related Parameter.

### 1.3 Theory:

#### 1.3.1 Introduction

Analysis of email is especially important not just because email may be used to communicate about things that we might be interested in for an investigation, but because it is a comparatively permanent and public record of those communications. In the case of a phone call, there is only the record that a call took place; in a spoken conversation, there may be no record at all. Conventional mail can be virtually untraceable, and paper documents are easily destroyed. Email, however, is unique; when a message is sent, the entire message is stored for both the sender and the receiver, and records of the mail being sent are stored on dozens of servers that the message passes through before arriving at its destination. There are a number of ways to analyze email, including: data mining techniques, which may be applied to large or small data sets; straightforward searching of a user's email for certain content; and in-depth analysis of an individual email's lineage.

E-mail system comprises of various hardware and software components that include sender’s client and server computers and receiver’s client and server computers with required software and services installed on each. Besides these, it uses various systems and services of the Internet. The sending and receiving servers are always connected to the

Internet but the sender's and receiver's client connects to the Internet as and when required.

An e-mail communication between a sender '*Alice'* having e-mail address '*alice@a.com*' and recipient '*Bob'* having e-mail address '*bob@b.com*' is shown in figure 1.'*Alice'* composes an e-mail message on her computer called client for '*Bob'* and sends it to her sending server 'smtp.a.org' using *SMTP* protocol. Sending server performs a lookup for the mail exchange record of receiving server '*b.org*' through Domain Name System (*DNS)* protocol on *DNS* server '*dns.b.org*'. The *DNS* server responds with the highest priority mail exchange server '*mx.b.org'* for the domain 'b.org'. Sending server establishes *SMTP* connection with the receiving server and delivers the e-mail message to the mailbox of '*Bob'* on the receiving server. '*Bob'* downloads the message from his mailbox on receiving server to local mailbox on his client computer using *POP3* or *IMAP* protocols. Optionally, '*Bob'* can also read the message stored in his server mailbox without downloading it to the local mailbox by using a Webmail program.

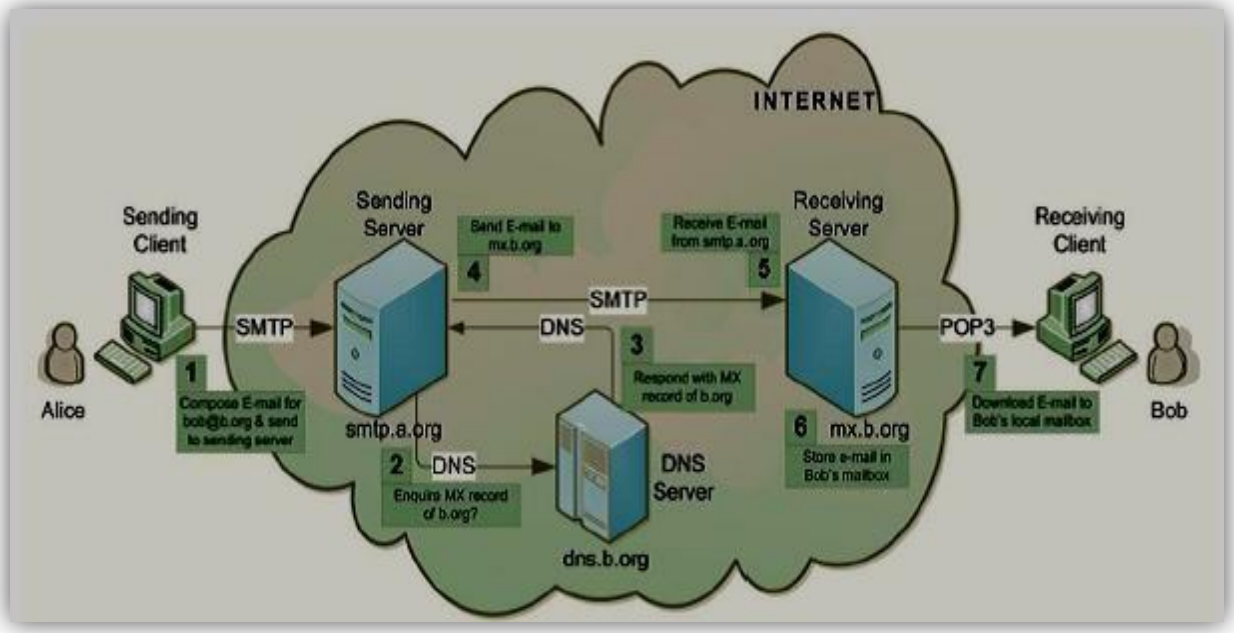



***Figure 1****: E-mail communication between a sender 'Alice' and recipient 'Bob'*

### 1.3.2 E-MAIL ACTORS, ROLES AND RESPONSIBILITIES

E-mail is a highly distributed service involving several actors that play different roles to accomplish end-to-end mail exchange. These actors fall under "User Actors", "Message Handling Service (*MHS*) Actors" and "Administrative Management Domain (*ADMD*) Actors" groups.

***User Actors*** are people, organizations or processes that serve as sources or sinks of messages. They can generate, modify or look at the whole message. User Actors can be of following four types (Table 1):

| User Actor Type | Roles and Responsibilities |
|---|---|
| **Author** | ▪ Responsible for creating the message, its contents, and its list of Recipient addresses.<br>▪ The MHS transfers the message from the Author and delivers it to the Recipients.<br>▪ The MHS has an Originator role that correlates with the Author role. |
| **Recipient** | ▪ The Recipient is a consumer of the delivered message.<br>▪ The MHS has a Receiver role that correlates with the Recipient role.<br>▪ A Recipient can close the user-communication loop by creating and submitting a new message that replies to the Author e.g. an automated form of reply is the Message Disposition Notification (MDN) |
| **Return Handler** | ▪ It is a special form of Recipient that provides notifications (failures or completions) generated by the MHS as it transfers or delivers the message.<br>▪ It is also called Bounce Handler. |
| **Mediator** | ▪ It receives, aggregates, reformulates, and redistributes messages among Authors and Recipients.<br>▪ It forwards a message through a re-posting process.<br>▪ It shares some functionality with basic MTA relaying, but has greater flexibility in both addressing and content than is available to MTAs. It preserves the integrity and tone of the original message, including the essential aspects of its origination information. It might also add commentary.<br>▪ It does not create new message that forwards an existing message, Reply or annotation.<br>▪ Some examples of mediators are: Alias, ReSender, Mailing Lists, Gateways and Boundary Filter. |

All types of Mediator user actors set HELO/EHLO, ENVID, RcptTo and Received fields. Alias actors also typically change To/CC/BCC and MailFrom fields. Identities relevant to ReSender are: From, Reply-To, Sender, To/CC/BCC, Resent-From, Resent-Sender, Resent-To/CC/BCC and MailFrom fields. Identities relevant to Mailing List processor are: List-Id, List-*, From, Reply-To, Sender, To/CC and MailFrom fields. Identities relevant to Gateways are: From, Reply-To, Sender, To/CC/BCC and MailFrom fileds.

***Message Handling Service (MHS) Actors*** are responsible for end-to-end transfer of messages. These Actors can generate, modify or look at only transfer data in the message. *MHS* Actors can be of following four types (Table 2):

| MHS Actor Type | Roles and Responsibilities |
|---|---|
| Originator | ▪ It ensures that a message is valid for posting and then submits it to a Relay<br>▪ It is responsible for the functions of the Mail Submission Agent.<br>▪ It also performs any post-submission that pertain to sending error and delivery notice.<br>▪ The Author creates the message, but the Originator handles any transmission issues with it |
| Relay | ▪ It performs MHS-level transfer-service routing and store-and-forward function by transmitting or retransmitting the message to its Recipients.<br>▪ It adds trace information but does not modify the envelope information or the semantics of message content.<br>▪ It can modify message content representation, such as changing the form of transfer encoding from binary to text, but only (as required) to meet the capabilities of the next hop in the MHS.<br>▪ When a Relay stops attempting to transfer a message, it becomes an Author because it sends an error message to the Return Address. |
| Gateway | ▪ It connects heterogeneous mail services despite differences in their syntax and semantics.<br>▪ It can send a useful message to a Recipient on the other side, without requiring changes to any components in the Author's or Recipient's mail services. |
| Receiver | ▪ It performs final delivery or sends the message to an alternate address.<br>▪ It can also perform filtering and other policy enforcement immediately before or after delivery. |

For networks, a port means an endpoint to a logical connection. The port number identifies what type (application/service offered) of port it is. The commonly used default port numbers used in e-mail are shown in Table 3. A complete list of default port numbering assignment is given in

| Port No | Protocols/Services | Description |
|---|---|---|
| 25 | SMTP<br>SMTP e-mail server | Simple Mail Transfer Protocol - core Internet protocol used to transfer from client to server (MUA to MTA) and server to server (MTA to MTA) |
| 110 | POP3<br>POP e-mail server | Post Office Protocol allows clients (MUA's) to retrieve stored e-mail |
| 143 | IMAP<br>IMAP(4) e-mail server | Internet Message Access Protocol provides a means of managing e-mail messages on a remote server and retrieve stored e-mail |
| 465 | SMTPS<br>WSMTP (SSMTP) protocol over TLS\|SSL | SMTP via SSL encrypted connection (Unofficial) |
| 993 | IMAPS<br>SSL encrypted IMAP | IMAP via SSL encrypted connection |
| 995 | POP3S SPOP<br>SSL encrypted POP | POP via SSL encrypted connection |
| 587 | MSA | Outgoing Mail (Submission) |
| 80 | HTTP | Webmail |
| 443 | HTTPS | Secure Webmail |

### 1.3.3 Analyzing an Individual Email

Although webmail will feature prominently in this section, the analysis of a particular email's lineage is much broader and can be applied to any email. A simple view of the path of an email from a sender to a client is presented in Figure 2. The email originates from the sender, whether from a local email client or a webmail application. When the email is sent, it is first sent to a Simple Mail Transfer Protocol (SMTP) server. That server forwards it to other SMTP servers until it finally reaches the destination server. On reaching its destination, the email is sent to a Post Office Protocol (POP) server, or any number of similar mail-delivery servers (IMAP is another, and webmail services may use their own servers for this purpose). The receiving client then connects to that server, retrieves the message, and allows the recipient to read it.

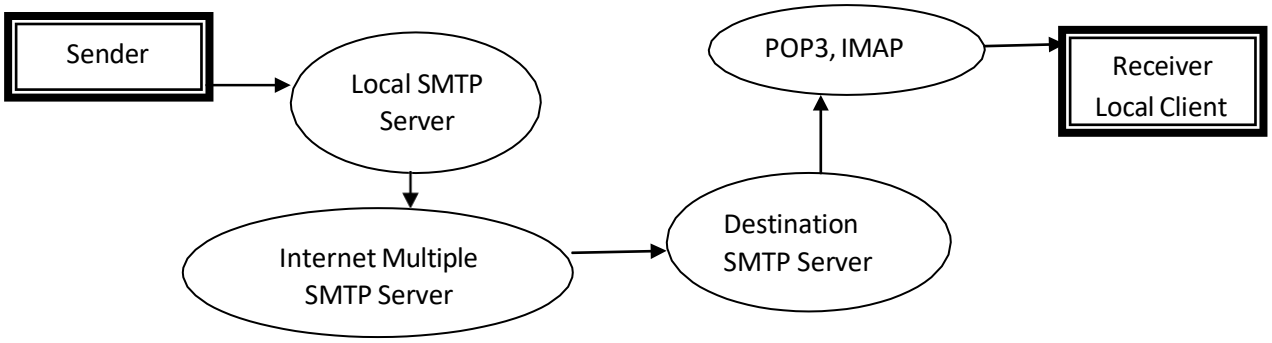



***Figure 2:*** *Path of an email from a sender to a client*

When the email is sent and when it is received, those respective servers add their own information to the email's header, and most likely log the action. Access to those logs may be required for much analysis, but specifics are outside of the scope of this paper. Considerable information can be gleaned from the header alone.

Suppose Moses, with the address moses@nmt.edu, sends an email from his office on the New Mexico Tech campus to his similarly named friend, with the email address thenewmoses@gmail.com. The subject of this email is "Snakes," and the content "Fish."

Below is the entire theoretical email, including all headers.

From:
moses@nmt.edu
Subject: Snakes
Date: September 25, 2021 9:35:29 PM
MDT To: thenewmoses@gmail.com
X-Gmail-Received:
ca493ed685a8e9ae77165ab2ce345127e5b310b4 Delivered-
To: thenewmoses@gmail.com
Received: by 10.90.33.15 with SMTP id g15cs279684agg; Mon, 25 Sep
2021 20:35:32 0700 (PDT)
Received: by 10.35.113.12 with SMTP id q12mr526602pym; Mon, 25 Sep 2021
20:35:32 -0700 (PDT)
Received: from mailhost.nmt.edu (mailhost.NMT.EDU
[129.138.4.52]) by mx.gmail.com with ESMTP id 36si2059018nza.
2021.09.25.20.35.32; Mon, 25 Sep 2021 20:35:32 -0700 (PDT)
Received: from localhost (localhost.localdomain [127.0.0.1]) by
localhost.localdomain (Postfix) with ESMTP id 09FF4436164 for
<thenewmoses@gmail.com>; Mon, 25 Sep 2021 21:35:32 -0600
(MDT) Received: from mailhost.nmt.edu ([127.0.0.1]) by localhost
(mailhost.nmt.edu [127.0.0.1]) (amavisd-new, port 10024) with
ESMTP id 11225-05 for <thenewmoses@gmail.com>; Mon, 25 Sep 2021
21:35:30 -0600 (MDT)
Received: from [192.168.1.2] (cs-fitch017.nmt.edu [129.138.21.110])
by mailhost.nmt.edu (Postfix) with ESMTP id 6FD4B436030 for
<thenewmoses@gmail.com>; Mon, 25 Sep 2021 21:35:30 -0600
(MDT) Return-Path: <moses@nmt.edu>
Received-Spf: pass (gmail.com: best guess record for
domain of moses@nmt.edu designates 129.138.4.52 as
permitted sender) Mime-Version: 1.0 (Apple Message
framework v752.2)
Content-Transfer-Encoding: 7bit
Message-Id: <77E313EF-271F-4AD0-A8D3-81263BF7B083@nmt.edu>

Content-Type: text/plain; charset=US-ASCII;
format=flowed X-Mailer: Apple Mail (2.752.2)
X-Virus-Scanned: by amavisd-new-2.3.1 (20050509) (RHEL AS) at
nmt.edu Fish

## E-MAIL IDENTITIES:

| Field Name | Set By | Field Description |
|---|---|---|
| **Layer:** ***Message Header Fields (Identification Fields)*** | | |
| Message- ID: | Originator | Globally unique message identification string generated when it is sent. |
| In-Reply-To: | Originator | Contains the Message-ID of the original message in response to which the reply message is sent. |
| References: | Originator | Identifies other documents related to this message, such as other e-mail message. |
| **Layer:** ***Message Header Fields (Originator Fields)*** | | |
| From: | Author | Name and e-mail address of the author of the message |
| Sender: | Originator | Contains the address responsible for sending the message on behalf of Author, if not omitted or same as that specified in From field. |
| Reply- To: | Author | E-mail address, the author would like recipients to use for replies. If present it overrides the From field. |
| **Layer:** ***Message Header Fields (Originator Date Fields)*** | | |
| Date: | Originator | It holds date and time when the message was made available for delivery. |
| **Layer:** ***Message Header Fields (Informational Fields)*** | | |
| Subject: | Author | It describes the subject or topic of the message. |
| Comments: | Author | It contains summarized comments regarding the message. |
| Keyword: | Author | It contains list of comma separated keywords that may be useful to the recipients e.g. when searching mail. |
| **Layer:** ***Message Header Fields (Destination Address Fields)*** | | |
| TO: | Author | Specifies a list of addresses of the recipients of the message. These addresses might be different from address in RcptTo SMTP commands |
| CC: | Author | Generally same as To Field. Generally a To field specifies primary recipient who is expected to take some action and CC addresses |

**1.4 Execution Steps**

- Open the Email which you want to analyze header
- Click on the right side three vertical dot(more) and select show original.
- New tab will be open then copy header information which you want to analyze.
- Open **https://www.whatismyip.com/email-header-analyzer** website.
- Copy the header information and click on analyze button.
- Then you will see the header analysis on screen.

**1.5 Assignment Question:**

1. Why to Analyze Email Header?
2. What Fields are analyzed during Email Analysis Header?
3. Which Readymade Tools are Available for Analyzing E-Mail Header?
4. Explain Email Architecture in Detail?
5. What is POP3, IMAP, SMTP, and MIME?

**1.6 Conclusion:**

E-mail is a widely used and highly distributed application involving several actors that play Different roles. These actors include hardware and software components, services and protocols which provide interoperability between its users and among the components along the path of transfer. Cybercriminals forge e-mail headers or send it anonymously for illegitimate purposes which lead to several crimes and thus make e-mail forensic investigation crucial.

**Assignment Group A-2**

**Problem Definition:**
**Implement a program to generate and verify CAPTCHA image.**

**2.1 Prerequisite:**

Basics of PYTHON

**2.2 Learning Objectives**:

1. Understand the use of CAPTCHA Image.
2. Generation and Verification of it.

**2.3New Concepts:**

1. CAPTCHA generation
2. Functions used like RANDOM

**2.4 Theory**

**2.4.1 Introduction**

A **CAPTCHA** (an acronym for "**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part") is a type of challenge-response test used in computing to determine whether or not the user is human. The most common type of CAPTCHA was first invented by Mark D. Lillibridge, Martin Abadi, Krishna Bharat and Andrei Z. Broder. This form of CAPTCHA requires that the user type the letters of a distorted image, sometimes with the addition of an obscured sequence of letters or digits that appears on the screen. Because the test is administered by a computer, in contrast to the standard Turing test that is administered by a human, a CAPTCHA is sometimes described as a reverse Turing test. Actually CAPTCHA is used as a simple puzzle hurdle, which restricts various automated

programs to sign-up E-mail accounts, cracking passwords, spam sending, privacy violation etc. This CAPTCHA actually challenges a particular automated program, which is trying to access some private zone. So, CAPTCHA helps in preventing access of personal mail accounts by some un-authorized automated spamming programs

### 2.4.2 Characteristics:-

CAPTCHAs are by definition fully automated, requiring little human maintenance or intervention to administer. This has obvious benefits in cost and reliability.

By definition, the algorithm used to create the CAPTCHA must be made public, though it may be covered by a patent. This is done to demonstrate that breaking it requires the solution to a difficult problem in the field of artificial intelligence (AI) rather than just the discovery of the (secret) algorithm, which could be obtained through reverse engineering or other means.

Modern text-based CAPTCHAS are designed such that they require the simultaneous use of three separate abilities—invariant recognition, segmentation, and parsing—to correctly complete the task with any consistency.

1. Invariant recognition refers to the ability to recognize the large amount of variation in the shapes of letters. There are nearly an infinite number of versions for each character that a human brain can successfully identify. The same is not true for a computer, and teaching it to recognize all those differing formations is an extremely challenging task.
2. Segmentation, or the ability to separate one letter from another, is also made difficult in CAPTCHAs, as characters are crowded together with no white space in between.
3. Context is also critical. The CAPTCHA must be understood holistically to correctly identify each character. For example, in one segment of a CAPTCHA, a letter might look like an “m.” Only when the whole word is taken into context does it become clear that it is a “u” and an “n.”

Each of these problems pose a significant challenge for a computer, even in isolation. The presence of all three at the same time is what makes CAPTCHAs difficult to solve.

### 2.4.3 Why we Prefer Captcha rather other security measures?

**1. To Protect Website's Registration Forms –**

Many Websites like Hotmail, Gmail, Yahoo, Facebook etc. offers free free registration. It is necessary to protect these website's registrations so that it ensures the registered user is a human not a program or bot. Captcha Code is used to protect the Registration Form Submission Programmatically.

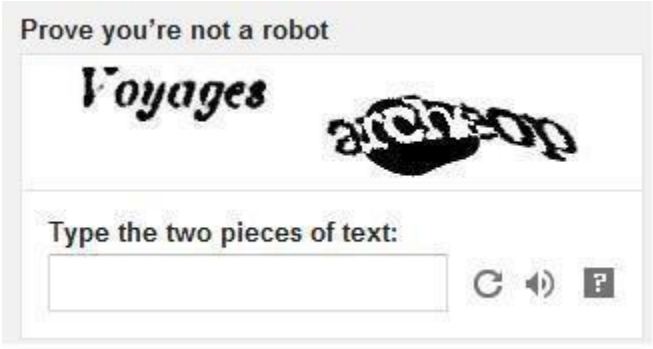



***Figure 1:-*** *Gmail Registration Form Captcha Code Image Screen Capture*

**2. To Prevent Comment Spams in Blogs –**

Captcha Code is used in the comment form so that only human can comment on a post otherwise spammers can flood hundreds of comments to a single post.

**3. To Protect Email Address Scrapping –**

Spammers crawl the web in the search of Email address posted in the clear text (e.g. **email@website.com**). You can protect your email address either by using Captcha to hide the email address, one can solve the Captcha before showing the Email address or by using alternative trick to post Email Address in the format of email at website dot com.

**4. To Protect from Search Engine Bots –**

Many Html tags are available to specifying indexing condition to Search engine bots. To prevent a website or specific webpage from search engine crawling, it is desirable to use

**html meta tag** but sometimes it is not completely sure that the webpage is fully protected from search engine crawlers and large companies who needs a high security uses Captcha rather than to use only meta tags to protect their highly public protected and confidential Web Pages.

### 2.4.4 Application of CAPTCHA:-

**Applications of CAPTCHAs**

CAPTCHAs have several applications for practical security:

- **Preventing Comment Spam in Blogs.** Most bloggers are familiar with programs that submit bogus comments, usually for the purpose of raising search engine ranks of some website (e.g., "buy penny stocks here"). This is called comment spam. By using a CAPTCHA, only humans can enter comments on a blog. There is no need to make users sign up before they enter a comment, and no legitimate comments are ever lost!
- **Protecting Website Registration.** Several companies (Yahoo!, Microsoft, etc.) offer free email services. Up until a few years ago, most of these services suffered from a specific type of attack: "bots" that would sign up for thousands of email accounts every minute. The solution to this problem was to use CAPTCHAs to ensure that only humans obtain free accounts. In general, free services should be protected with a CAPTCHA in order to prevent abuse by automated scripts.
- **Protecting Email Addresses From Scrapers.** Spammers crawl the Web in search of email addresses posted in clear text. CAPTCHAs provide an effective mechanism to hide your email address from Web scrapers. The idea is to require users to solve a CAPTCHA before showing your email address.
- **Online Polls.** In November 1999, *http://www.slashdot.org* released an online poll asking which was the best graduate school in computer science (a dangerous question to ask over the web!). As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once. However, students at Carnegie Mellon found a way to stuff the ballots using

programs that voted for CMU thousands of times. CMU's score started growing rapidly. The next day, students at MIT wrote their own program and the poll became a contest between voting "bots." MIT finished with 21,156 votes, Carnegie Mellon with 21,032 and every other school with less than 1,000. Can the result of any online poll be trusted? Not unless the poll ensures that only humans can vote.

- **Preventing Dictionary Attacks.** CAPTCHAs can also be used to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins. This is better than the classic approach of locking an account after a sequence of unsuccessful logins, since doing so allows an attacker to lock accounts at will.
- **Search Engine Bots.** It is sometimes desirable to keep WebPages unindexed to prevent others from finding them easily. There is an html tag to prevent search engine bots from reading web pages. The tag, however, doesn't guarantee that bots won't read a web page; it only serves to say "no bots, please." Search engine bots, since they usually belong to large companies, respect web pages that don't want to allow them in. However, in order to truly guarantee that bots won't enter a web site, CAPTCHAs are needed.
- **Worms and Spam.** CAPTCHAs also offer a plausible solution against email worms and spam: "I will only accept an email if I know there is a human behind the other computer." A few companies are already marketing this idea.

**2.4.5 Advantages:**

1. Distinguishes between a human and a machine
2. Makes online polls more legitimate
3. Reduces spam and viruses
4. Makes online shopping safer
5. Diminishes abuse of free email account services

**2.4.6 Disadvantages:**

1. Sometimes very difficult to read
2. Are not compatible with users with disabilities
3. Time-consuming to decipher
4. Technical difficulties with certain internet browsers
5. May greatly enhance Artificial Intelligence

**2.5 Algorithm:**

1. Start
2. Import **image.captcha** from ImageCaptcha
3. Generate captcha by **generate(captcha_text)**
4. For randomly captcha generation use function **random.randint()**
5. Create an image to write a captcha text
6. End

**5.7 Assignment Questions:**

1. What is Full Form of CAPTCHA?
2. Write down different forms of CAPTCHA?
3. Why CAPTCHA is needed?
4. Explain the uses of different captcha as per Requirement?

**Conclusion:**

Hence we conclude that CAPTCHA is used to distinguished Human and Machine and Provide Security to Programs.

**Assignment Group A-3**

**Problem Definition: Write a Computer Forensics Application Program in Java/Python/C++ for recovering Deleted Files and Deleted Partitions.**

### 3.1 Prerequisite:

a) Knowledge about Partitions in Ubuntu. b) Path of Trash folder.

### 3.2 Learning Objectives:

- Understand the concept of Recovery of deleted files.
- Implementation of recovery of deleted Partitions.

### 3.3 New Concepts:

a. Recovery of files in LINUX OS.

### 3.4 Theory

#### 3.4.1 Introduction

Have you accidentally deleted an important file because you are in a habit of using "Shift+Del" rather than delete only?? Well don't panic.There are many utilities in Ubuntu and other Linux distributions which helps you in recovering the so called "permanently deleted" files. Actually when you delete a file permanently (accidentally or intentionally), It doesn't get removed from your hard disk.It get stored in certain blocks of the storage device and they continue to exist in the blocks unless you overwrite them with newer files. There are many Tools available to recover permanently deleted files Scalpel.

Scalpel is a platform independent command based tool which is small yet very powerful. But, if the file is deleted i.e. by just pressing Delete button the file is stored in Trash folder in Ubuntu OS. So it is easy to recover the deleted files from Trash Folder. Just we need to know the path of trash folder.

**Path is: ="/home/gurukul/.local/share/Trash/files"**

There are sub-Folders in Trash Folder namely :

1. files- contains files which are deleted
2. info- contains information of files deleted
3. expunged

### 3.4.2 Introduction to file systems:

File systems are one of the things any newcomer to linux must become acquainted with. In the world of Microsoft you never really have to worry about it, the default being NTFS. Linux however, being built on a world of open source and differing opinions, is not limited in this way and so the user should have an understanding of what a file system is, and how it affects the computer.

At the core of a computer, it's all 1s and 0s, but the organization of that data is not quite as simple. A *bit* is a 1 or a 0, a *byte* is composed of 8 bits, a kilobyte is 1024 (i.e. 2) bytes, a megabyte is 1024 kilobytes and so on and so forth. All these *bits* and *bytes* are permanently stored on a Hard Drive. A hard drive stores all your data, any time you save a file, you're writing thousands of 1s and 0s to a metallic disc, changing the magnetic properties that can later be read as 1 or 0. There is so much data on a hard drive that there has to be some way to organize it, like a library of books and the old card drawers that indexed all of them, without that index, we'd be lost. Libraries, for the most part, use the Dewey Decimal System to organize their books, but

there exist other systems to do so, none of which have attained the same fame as Mr. Dewey's invention. File systems are the same way. The ones most users are aware of are the ones Windows uses, the vFat or the NTFS systems, these are the Windows default file systems.

Ubuntu (like all UNIX-like systems) organizes files in a hierarchical tree, where relationships are thought of in teams of children and parent. *Directories* can contain other directories as well as *regular files*, which are the "leaves" of the tree. Any element of the tree can be references by a *path name*; an *absolute path name* starts with the character / (identifying the *root directory*, which contains all other directories and files), then every child directory that must be traversed to reach the element is listed, each separated by a / sign.

### 3.4.3 Main directories

The standard Ubuntu directory structure mostly follows the File system Hierarchy Standard, which can be referred to for more detailed information.

Here, only the most important directories in the system will be presented.
**/bin** is a place for most commonly used terminal commands, like ls, mount, rm, etc.

**/boot** contains files needed to start up the system, including the Linux kernel, a RAM disk image and bootloader configuration files.

**/dev** contains all *device files*, which are not regular files but instead refer to various hardware devices on the system, including hard drives.

**/etc** contains system-global configuration files, which affect the system's behavior for all users. **/home** home sweet home, this is the place for users' home directories.

**/lib** contains very important dynamic libraries and kernel modules

**/media** is intended as a mount point for external devices, such as hard drives or removable media (floppies, CDs, DVDs).

**/mnt** is also a place for mount points, but dedicated specifically to "temporarily mounted" devices, such as network filesystems.

**/opt** can be used to store addition software for your system, which is not handled by the package manager.

**/proc** is a virtual filesystem that provides a mechanism for kernel to send information to processes.

**/root** is the superuser's home directory, not in /home/ to allow for booting the system even if /home/ is not available.

**/sbin** contains important administrative commands that should generally only be employed by the superuser.

**/srv** can contain data directories of services such as HTTP (/srv/www/) or FTP.

**/sys** is a virtual filesystem that can be accessed to set or obtain information about the kernel's view of the system.

**/tmp** is a place for temporary files used by applications.

**/usr** contains the majority of user utilities and applications, and partly replicates the root

directory structure, containing for instance, among others, /usr/bin/ and /usr/lib.

**/var** is dedicated variable data that potentially changes rapidly; a notable directory it contains is /var/log where system log files are kept. **Steps to Partition HardDisk Drive in Ubuntu:-**

**Step 1**. If you are trying to format or partition your hard drive it is assumed that bios is able to detect the device. To determine the path and other specific information about your drive open a terminal window and enter this command:

sudo lshw -C disk

**Step 2.** After entering this command Ubuntu should return something similar to this. Take note of the "logical name" because this will be used throughout the partitioning process if done via terminal window**.**

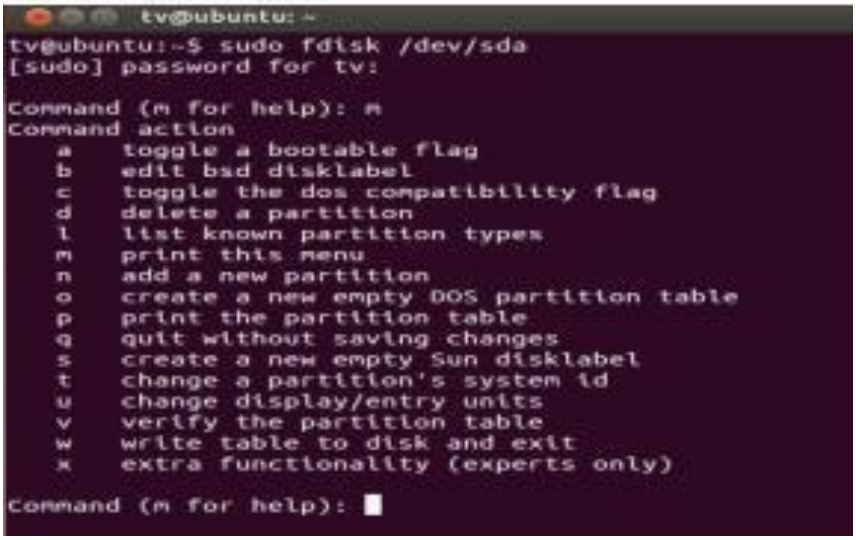

**Step 3.** The part we will be most concerned with will be the hard drive information that is displayed in the terminal window.

If you plan on using the hard drive only for Ubuntu then the recommended file system to use is either ext3/ext4 depending on whether or not you need backwards compatibility with previous versions of Linux. If you will need to share files between Ubuntu and Windows machines fat 32 is the recommended file system to use, but NTFS will also work well also.

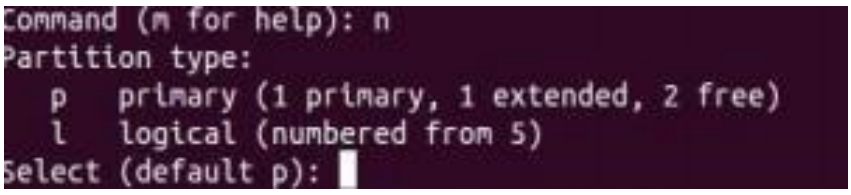
### 3.4.4 Partition using command line in Terminal: Step

**1.** Start fdisk with this command

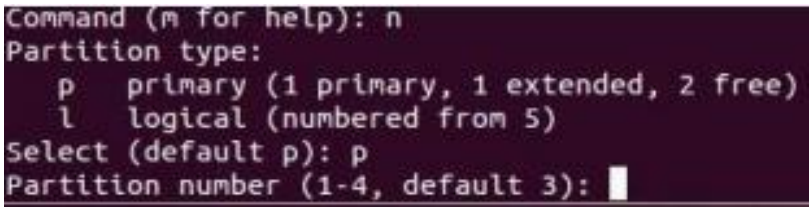

**Step 2.** Press **"m"** then hit **enter**. This will return a menu like the one below showing all of the available commands for the fdisk program.




**Step 3.** Since we want to add a new partition press **"n"** and then **enter**.




**Step 4.** To create a primary partition (what we want) press **"p"** and then hit **enter**.



**Step 5.**

If you only want 1 partition press **"1"** and hit **enter**. You may be provided with a default response, you may choose this as the Partition number if you would like. Next you will be prompted for the locations of where you would like the first and last sectors of the partition to be. You may again be provided with default responses choose these if you want.

**Step 6.**

Now choose w to write the partition to the disk. Type "w" then press enter.Your drive is now partitioned. Now we need to format it. By default Linux will recognize this partition as dev/sdb1.

**Step 7.** To format the partition with an ext3 filesystem.

**sudo mkfs -t ext3 /dev/sdb1**

**3.5 Algorithm:**

1. Start
2. Initialize variables as path="/home/gurukul/.local/share/Trash/files" infopath="/home/gurukul/.local/share/Trash/info"
3. Check the list of files present in files folder.
4. Find the path of file to restore it using info folder.
5. Copy the contents of file which is deleted and is in Trash folder into new file at original location.
6. Delete the file from Trash folder.
7. End

**3.6 Mathematical Model:**

I= P (path of trash folder) **Functions:**
re.findall(r'/.*',line)
destipath.lstrip('[') destipath.rstrip(']')
destipath[:-1]
destipath[1:] **Output:**
R- Recovered file

**3.7 Assignment Questions:**

1. What is Path of Trash folder in Ubuntu and what are the different folders?
2. How to see hidden files and filesystem of ubuntu?
3. What are different file systems in Ubuntu also state main directories of it??
4. How to list different files and what are the various options of ls used for file related function?

**Conclusion:**

Hence we conclude that using **Forensics Application Program in Python we can recover Deleted Files.**

**Assignment Group A-4**

**Write a program for Log Capturing and Event Correlation**

**Prerequisite:**

- Latest version of Squid should be used.(version 2.5 or greater)
- A web server for testing purpose which can be used instead of Internet.
- Squid Version greater than 2.6 is required for Transparent squid proxy configuration in this lab.
-

**Learning Objectives:**

- To understand how Log Records are generated for Further Analysis.

**New Concepts:**

- Squid and Sarg

**Theory**

**4.1. Introduction:**

- During the period of development of internet, users are allowed for unlimited access to the resources due to less number of users. So there were less issues related to accessing speed over internet.
- With the increase in internet usage, many issues raised related to accessing speed, effective bandwidth utilization etc. One method of overcoming these issues is, maintaining a copy of webpage visited by a user in the cache so that the other user who visits the same webpage will access the same website within a short period of time. This method not only increases the accessing speed but also helps in utilizing the bandwidth effectively.
- The above said functionality can be achieved by maintaining a proxy server through which all the users in the organization or a group access the internet. The most widely used proxy server in Linux is Squid Proxy, which is free software released General Public License.

- Squid provides proxy and cache services for **H**yper **T**ext **T**ransfer **P**rotocol (HTTP),

**F**ile **T**ransfer **P**rotocol (FTP), and various other protocols.

To configure a system as a proxy server, one should have a sufficient amount of memory for maintaining the cache which in turn increases the performance.

- In case if the internet connection is not available, setup one host as a web server in place of internet and assign the IP address to the proxy server network interface in the network, used by web server instead of public IP address assigned to that interface.

## 4.2. Steps to Configure Squid Proxy:

### 4.2.1. Installation of Squid Package

A Squid proxy server is generally installed on a separate server than the Web server with the original files. Squid works by tracking object use over the network. Squid will initially act as an intermediary, simply passing the client's request on to the server and saving a copy of the requested object. If the same client or multiple clients request the same object before it expires from Squid's cache, Squid can then immediately serve it, accelerating the download and saving bandwidth.

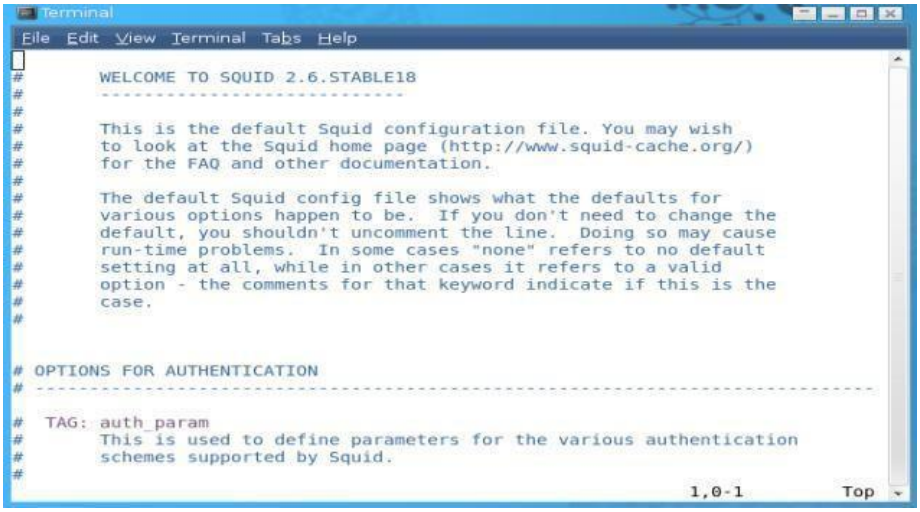
```
sudo apt update
sudo apt -y install squid
```

### 4.2.2. Accessing the Proxy Server configuration file

To configure squid proxy server we need to edit the *sudo gedit /etc/squid/squid.conf* file and the default location of squid.conf file varies from distribution to distribution and from version to version. We can edit the configuration file using vi editor through command prompt.

```
sudo gedit /etc/squid/squid.conf
```

Then the content of the configuration file can be viewed as shown below in the figure.




**Editing the squid configuration file**

**sudo gedit /etc/squid/squid.conf**

Search the TAG: auth_param and paste the following acl

```
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 2 hours
acl auth_users proxy_auth REQUIRED
http_access allow auth_users

 #search localnet and paste line in last

acl localnet src 192.168.60.70
```

>sudo service squid3 restart

**Specifying the interface and port number on which the proxy server should listen.**

By default, the proxy server will listen on all the available network interfaces on the system for requests. For Example, if one interface card is assigned a public ip from which it is connected to internet and the other interface card is assigned an ip address which belongs to your local area network. Then in order to make you proxy server to listen for requests from your Local Area Network through a particular port, then change the variable http_port 3128 in the squid configuration file to desired ip address and port number in the format shown below.

http_port <ip address belonging to LAN>:<port number>

Example: For example, if your proxy server has an ip address 192.168.60.70 which belongs to the local area network 192.168.60.0/24 and you want the server to listen for requests from your LAN through a particular port say 3456, then you can change the variable http_port as shown.

http_port 192.168.60.70:3456

**Assigning Access Controls**

By default, no user machine is allowed to connect to the proxy server except the localhost. To allow the local machines access your proxy server, locate the acl section in the squid configuration file starting with acl and at the end of the last acl line specify your access

control. For example to allow local area network 192.168.60.0/24 machines to access your proxy server, specify the acl as

acl mylan src 192.168.60.0/255.255.255.0

In the above example, mylan specifies the name of my access control. We can specify any name other than my lan for access control. src specifies the source network.

**Allow or Deny based on Access Control**.

After specifying the access control for your local LAN, we need to provide allow permission for the specified LAN using http_access variable in the squid configuration file as shown in the example below.

Example: To allow the above specified access control ( i.e acl mylan src 192.168.60.0/255.255.255.0), we need to specify the http_access variable as



Hyderabad http_access allow mylan

Here mylan specifies the access control used. Suppose if we want to allow all the networks except the 192.168.60.0/24 network to access the proxy then we can specify the http_access variable as

http_access deny !mylan

In the above line, !mylan specifies except mylan network.

**Note:**

The above specified http_access variable should be specified before the line

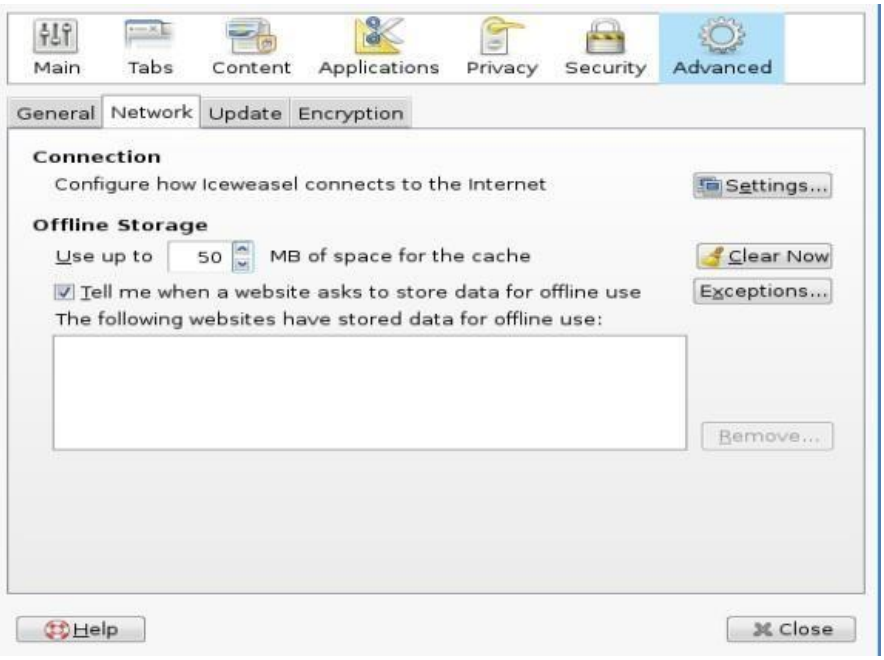
http_access deny all in the configuration file.

**Saving the changes and exit the gedit Editor**

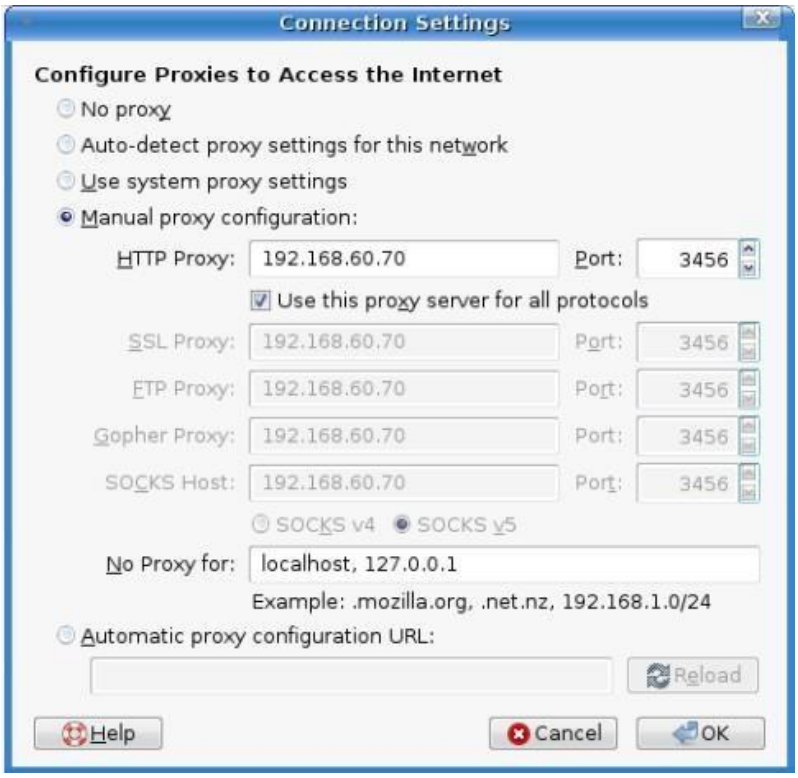
After making appropriate changes to your configuration file exit the vi editor window by pressing Esc followed by :wq!. Here wq specifies save changes and exit the configuration file.

### 4.3 Testing the Squid configuration

To test the squid configuration, open a browser in any one of the pc in local area network or on the proxy server and specify the proxy settings as the ipaddress of the proxy server and port on which it is listening for requests. For example, in firefox web browser if we want to set the proxy settings in the browser window goto **Edit** -->**Preferences** and window similar to shown below will be displayed.




Now select Advanced tab, and under advanced tab click on Network tab and click on Settings option under Connection field. Then a window similar to the shown below will be displayed.




## 4.4 SARG – Squid Analysis Report Generator and Internet Bandwidth Monitoring Tool

SARG is an open source tool that allows you to analyze the squid log files and generates
beautiful reports in HTML format with information's about users, IP addresses, top accessed sites, total bandwidth usage, elapsed time, downloads, access denied websites, daily reports, weekly reports and monthly reports.

The SARG is very handy tool to view how much internet bandwidth is utilized by individual machines on the network and can watch on which websites the network's users are accessing.

Installing Sarg from Source

The '**sarg**'package by default not included in **RedHat** based distributions, so we need to manually compile and install it from source tar ball. For this, we need some additional pre- requisites packages to be installed on the system before compiling it from source.

**$ sudo apt-get install sarg**

Configuring Sarg

Now it's time to edit some parameters in SARG main configuration file. The file contains lots of options to edit, but we will only edit required parameters like:

Access logs

path

Output

directory

Date

Format

Overwrite report for the same date.

**Open sarg.conf file with your choice of editor and make changes as shown below.**

# vi /usr/local/etc/sarg.conf [On RedHat based systems]

Now Uncomment and add the original path to your squid access
log file. # sarg.conf

# TAG: access_log file
# Where is the access.log file
# sarg
-l file
access_log /var/log/squid/access.log

Next, add the correct Output directory path to save the generate squid reports in that directory. Please note, under Debian based distributions the Apache web root directory is '/var/www'. So, please be careful while adding correct web root paths under your Linux distributions.

# TAG: output_dir
# The reports will be saved in that
directory # sarg -o dir

output_dir /var/www/html/squid-reports

**Set the correct date format for reports. For example, 'date_format e' will**

**display reports in 'dd/mm/yy' format.**

# TAG: date_format

# Date format in reports: e (European=dd/mm/yy), u (American=mm/dd/yy), w (Weekly=yy.ww)

#

date_format e

Next, uncomment and set Overwrite report to 'Yes'. # TAG: overwrite_report yes|no

# yes - if report date already exist then will be overwritten.

# no - if report date already exist then will be renamed to filename.n, filename.n+1 #

overwrite_report yes

That's it! Save and close the file.

**Step 3: Generating Sarg Report**

Once, you've done with the configuration part, it's time to generate the squid log report using the following command.

# sarg -x [On RedHat based systems]

**Assessing Sarg Report**

The generated reports placed under '/var/www/html/squid-reports/' or '/var/www/squid-reports/' which can be accessed from the web browser using the address.

http://localhost/squid-reports OR

http://ip-address/squid-reports

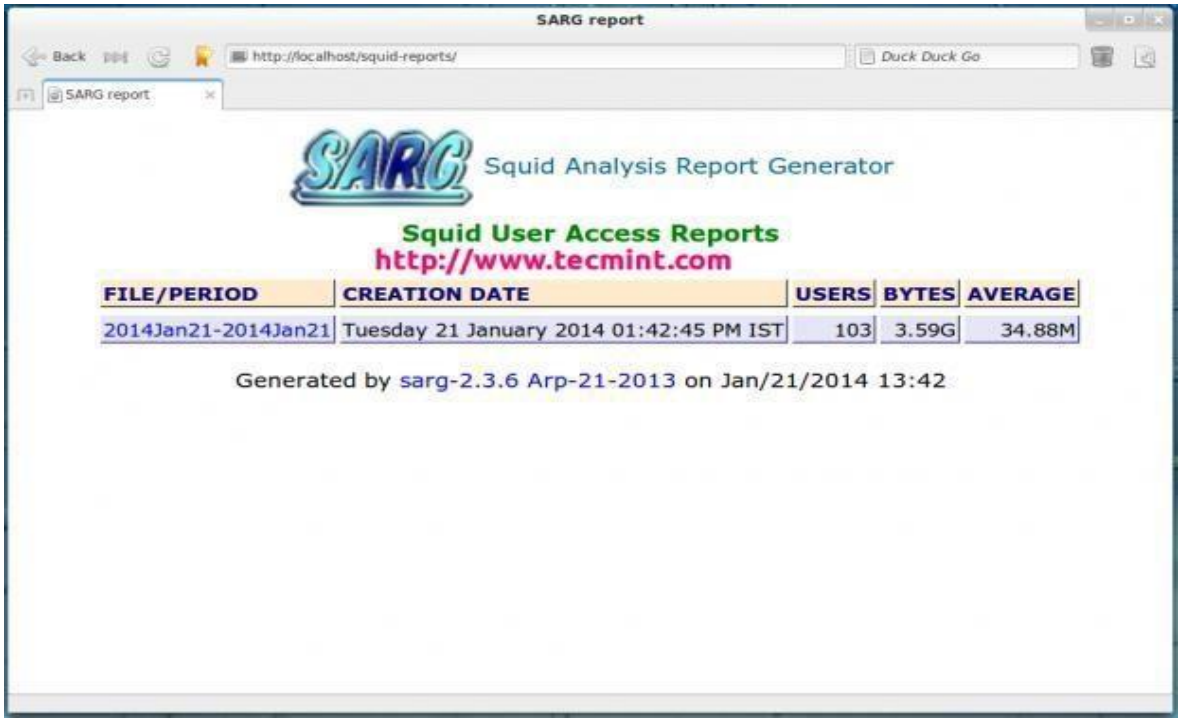



**Fig.1 Sarg Main Window**

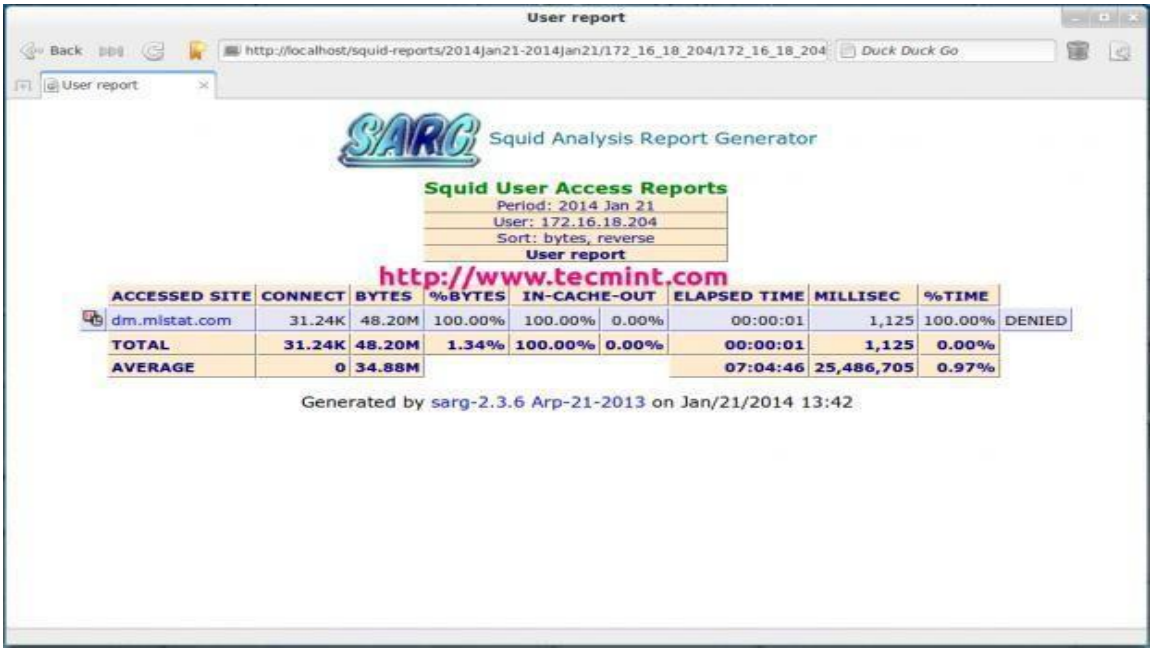
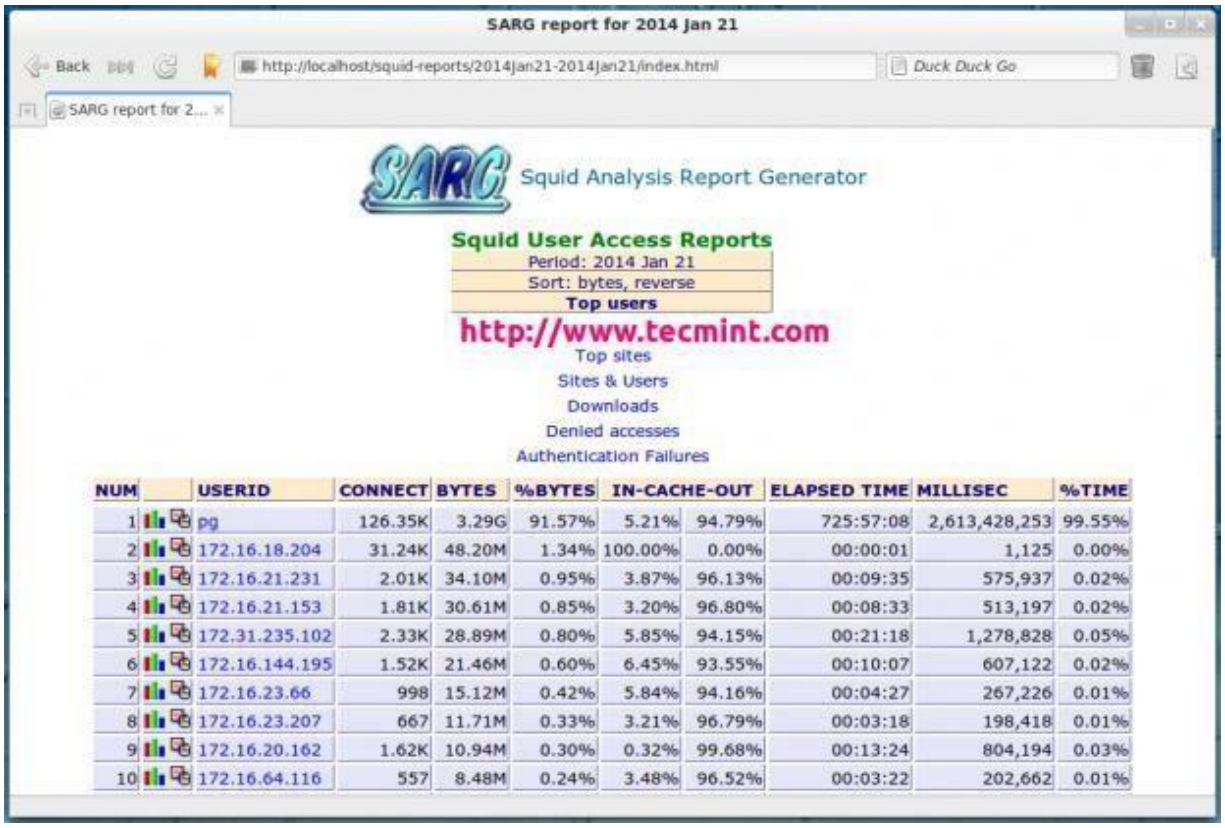
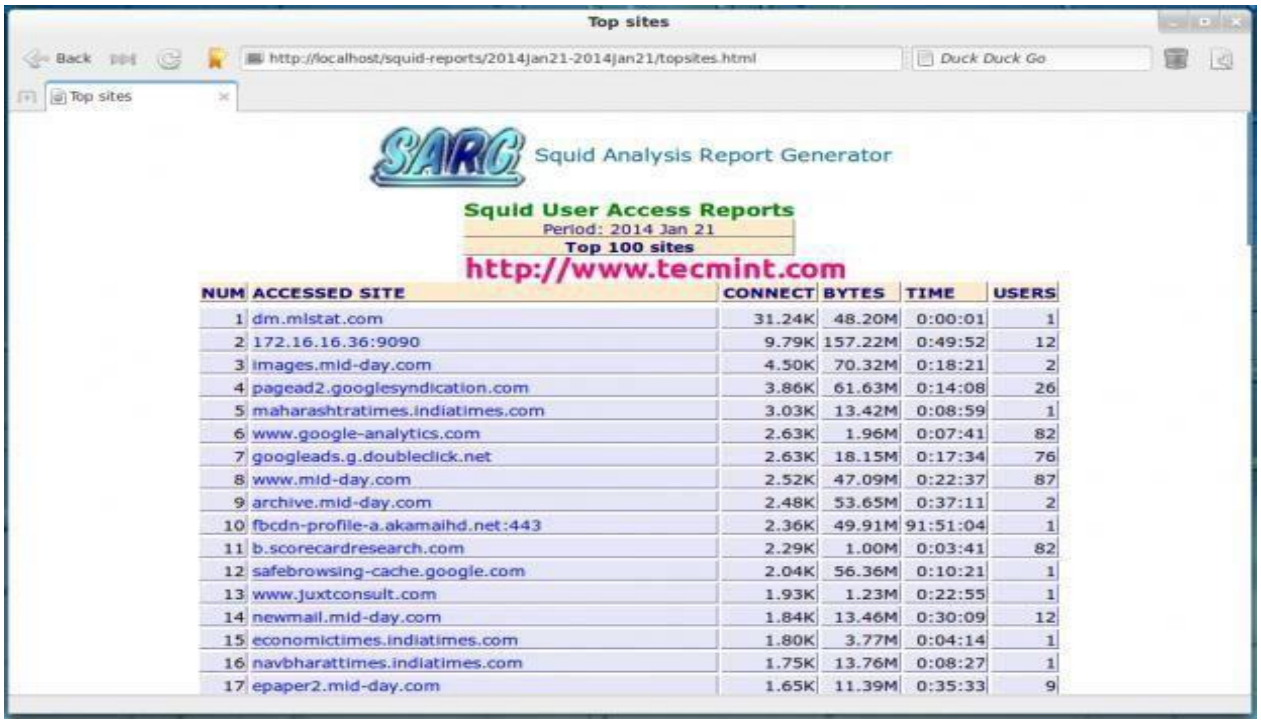



**Fig2.User Report**



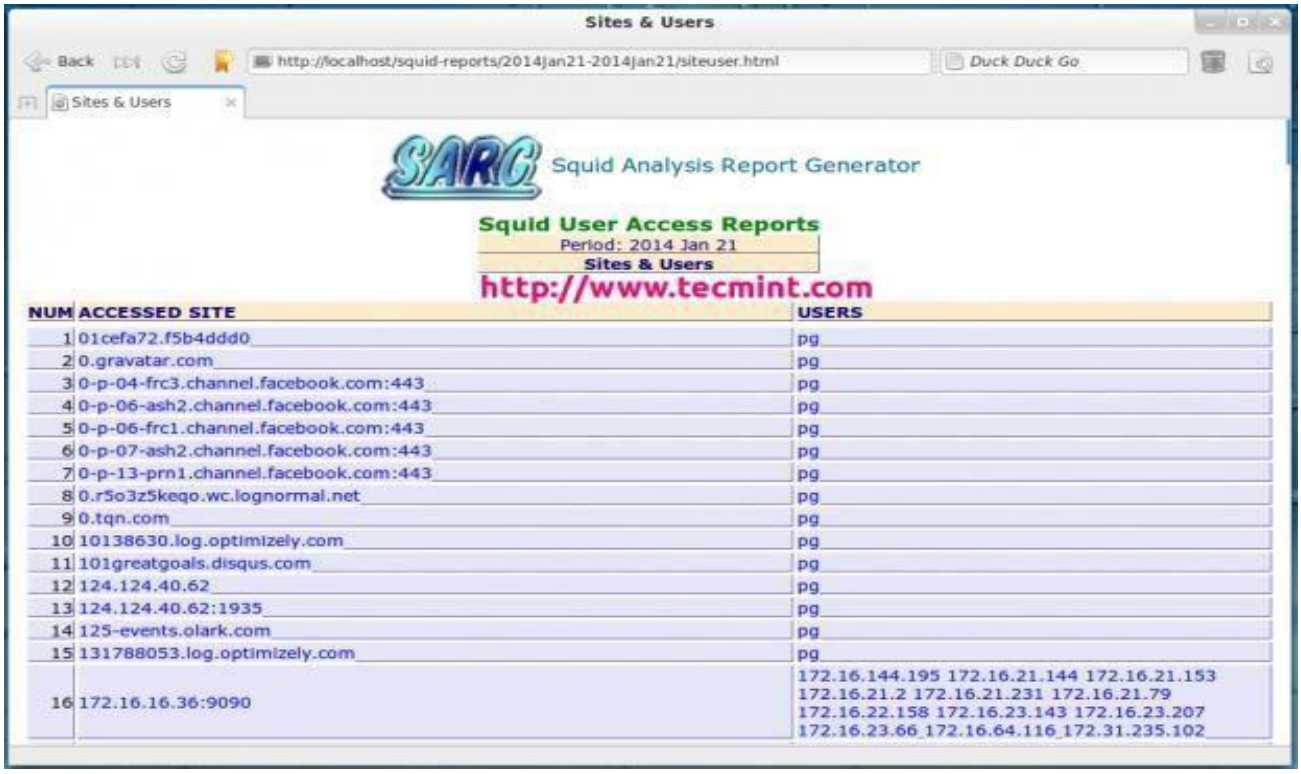

| NUM | USERID | CONNECT | BYTES | %BYTES | IN-CACHE-OUT | | ELAPSED TIME | MILLISEC | %TIME |
|---|---|---|---|---|---|---|---|---|---|
| 1 | pg | 126.35K | 3.29G | 91.57% | 5.21% | 94.79% | 725:57:08 | 2,613,428,253 | 99.55% |
| 2 | 172.16.18.204 | 31.24K | 48.20M | 1.34% | 100.00% | 0.00% | 00:00:01 | 1,125 | 0.00% |
| 3 | 172.16.21.231 | 2.01K | 34.10M | 0.95% | 3.87% | 96.13% | 00:09:35 | 575,937 | 0.02% |
| 4 | 172.16.21.153 | 1.81K | 30.61M | 0.85% | 3.20% | 96.80% | 00:08:33 | 513,197 | 0.02% |
| 5 | 172.31.235.102 | 2.33K | 28.89M | 0.80% | 5.85% | 94.15% | 00:21:18 | 1,278,828 | 0.05% |
| 6 | 172.16.144.195 | 1.52K | 21.46M | 0.60% | 6.45% | 93.55% | 00:10:07 | 607,122 | 0.02% |
| 7 | 172.16.23.66 | 998 | 15.12M | 0.42% | 5.84% | 94.16% | 00:04:27 | 267,226 | 0.01% |
| 8 | 172.16.23.207 | 667 | 11.71M | 0.33% | 3.21% | 96.79% | 00:03:18 | 198,418 | 0.01% |
| 9 | 172.16.20.162 | 1.62K | 10.94M | 0.30% | 0.32% | 99.68% | 00:13:24 | 804,194 | 0.03% |
| 10 | 172.16.64.116 | 557 | 8.48M | 0.24% | 3.48% | 96.52% | 00:03:22 | 202,662 | 0.01% |

**Fig 3. Specific Date**



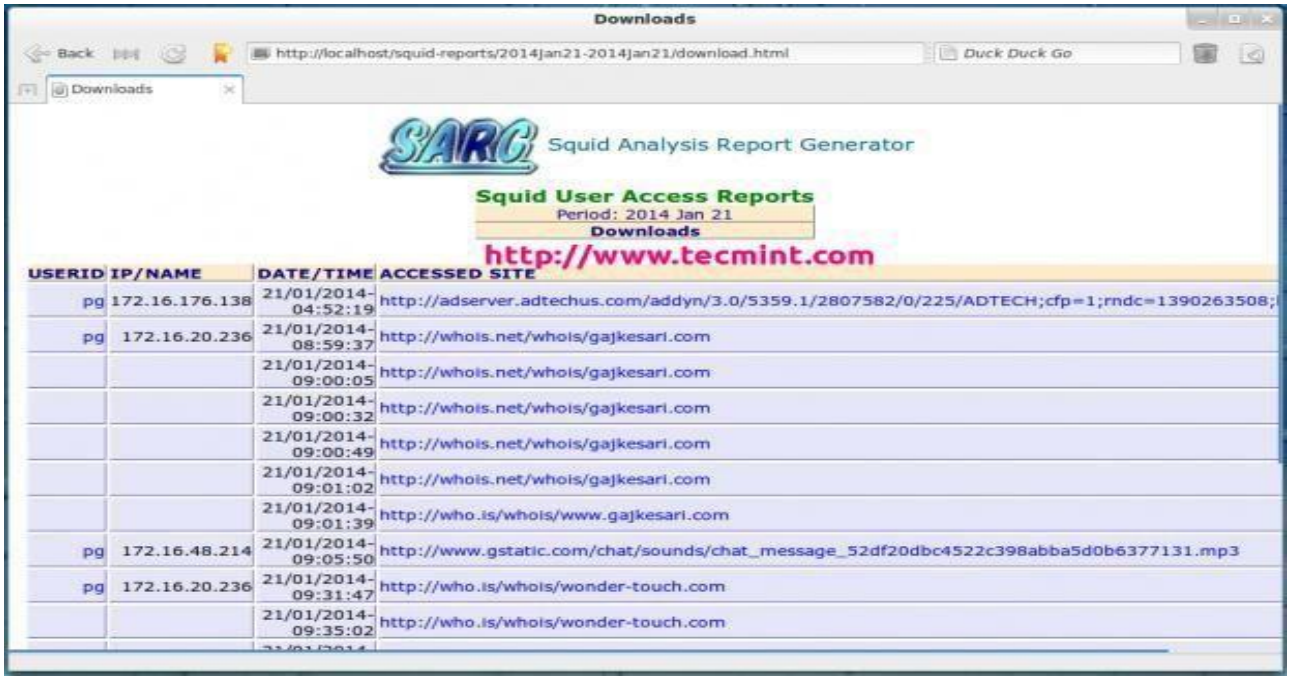

| NUM | ACCESSED SITE | CONNECT | BYTES | TIME | USERS |
|---|---|---|---|---|---|
| 1 | dm.mistat.com | 31.24K | 48.20M | 0:00:01 | 1 |
| 2 | 172.16.16.36:9090 | 9.79K | 157.22M | 0:49:52 | 12 |
| 3 | images.mid-day.com | 4.50K | 70.32M | 0:18:21 | 2 |
| 4 | pagead2.googlesyndication.com | 3.86K | 61.63M | 0:14:08 | 26 |
| 5 | maharashtratimes.indiatimes.com | 3.03K | 13.42M | 0:08:59 | 1 |
| 6 | www.google-analytics.com | 2.63K | 1.96M | 0:07:41 | 82 |
| 7 | googleads.g.doubleclick.net | 2.63K | 18.15M | 0:17:34 | 76 |
| 8 | www.mid-day.com | 2.52K | 47.09M | 0:22:37 | 87 |
| 9 | archive.mid-day.com | 2.48K | 53.65M | 0:37:11 | 2 |
| 10 | fbcdn-profile-a.akamaihd.net:443 | 2.36K | 49.91M | 91:51:04 | 1 |
| 11 | b.scorecardresearch.com | 2.29K | 1.00M | 0:03:41 | 82 |
| 12 | safebrowsing-cache.google.com | 2.04K | 56.36M | 0:10:21 | 1 |
| 13 | www.juxtconsult.com | 1.93K | 1.23M | 0:22:55 | 1 |
| 14 | newmail.mid-day.com | 1.84K | 13.46M | 0:30:09 | 12 |
| 15 | economictimes.indiatimes.com | 1.80K | 3.77M | 0:04:14 | 1 |
| 16 | navbharattimes.indiatimes.com | 1.75K | 13.76M | 0:08:27 | 1 |
| 17 | epaper2.mid-day.com | 1.65K | 11.39M | 0:35:33 | 9 |

**Fig 4. Top Accessed Sites**



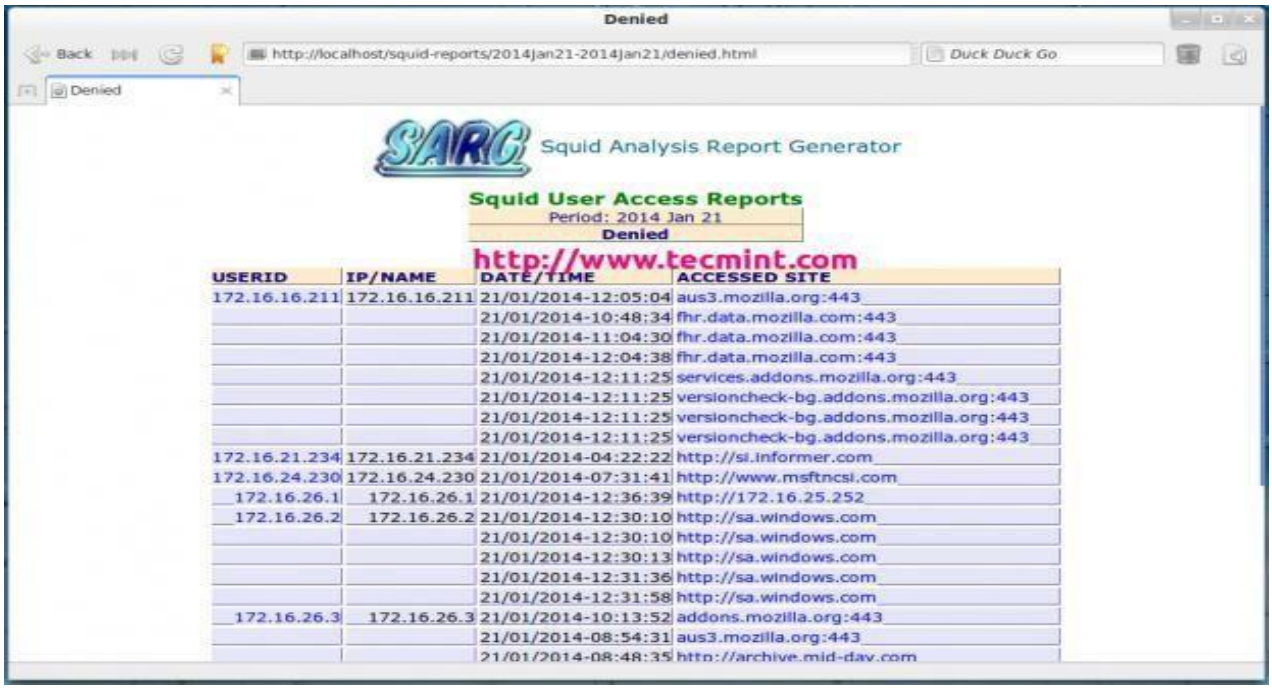

| NUM | ACCESSED SITE | USERS |
|---|---|---|
| 1 | 01cefa72.f5b4ddd0 | pg |
| 2 | 0.gravatar.com | pg |
| 3 | 0-p-04-frc3.channel.facebook.com:443 | pg |
| 4 | 0-p-06-ash2.channel.facebook.com:443 | pg |
| 5 | 0-p-06-frc1.channel.facebook.com:443 | pg |
| 6 | 0-p-07-ash2.channel.facebook.com:443 | pg |
| 7 | 0-p-13-prn1.channel.facebook.com:443 | pg |
| 8 | 0.r5o3z5keqo.wc.lognormal.net | pg |
| 9 | 0.tqn.com | pg |
| 10 | 10138630.log.optimizely.com | pg |
| 11 | 101greatgoals.disqus.com | pg |
| 12 | 124.124.40.62 | pg |
| 13 | 124.124.40.62:1935 | pg |
| 14 | 125-events.olark.com | pg |
| 15 | 131788053.log.optimizely.com | pg |
| 16 | 172.16.16.36:9090 | 172.16.144.195 172.16.21.144 172.16.21.153 172.16.21.2 172.16.21.231 172.16.21.79 172.16.22.158 172.16.23.143 172.16.23.207 172.16.23.66 172.16.64.116 172.31.235.102 |

**Fig 5. Top Sites and Users**



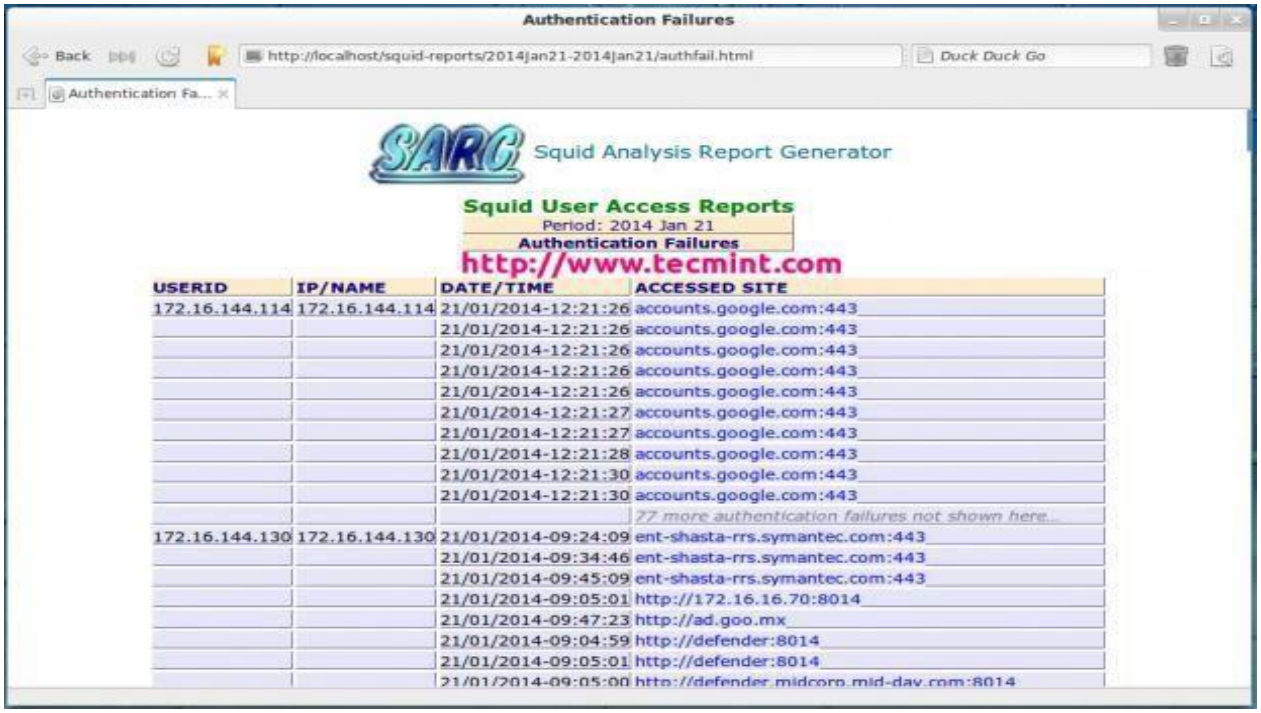

| USERID | IP/NAME | DATE/TIME | ACCESSED SITE |
|---|---|---|---|
| pg | 172.16.176.138 | 21/01/2014-04:52:19 | http://adserver.adtechus.com/addyn/3.0/5359.1/2807582/0/225/ADTECH;cfp=1;rndc=1390263508; |
| pg | 172.16.20.236 | 21/01/2014-08:59:37 | http://whois.net/whois/gajkesari.com |
| | | 21/01/2014-09:00:05 | http://whois.net/whois/gajkesari.com |
| | | 21/01/2014-09:00:32 | http://whois.net/whois/gajkesari.com |
| | | 21/01/2014-09:00:49 | http://whois.net/whois/gajkesari.com |
| | | 21/01/2014-09:01:02 | http://whois.net/whois/gajkesari.com |
| | | 21/01/2014-09:01:39 | http://who.is/whois/www.gajkesari.com |
| pg | 172.16.48.214 | 21/01/2014-09:05:50 | http://www.gstatic.com/chat/sounds/chat_message_52df20dbc4522c398abba5d0b6377131.mp3 |
| pg | 172.16.20.236 | 21/01/2014-09:31:47 | http://who.is/whois/wonder-touch.com |
| | | 21/01/2014-09:35:02 | http://who.is/whois/wonder-touch.com |

**Fig 6. Top Downloads**




| USERID | IP/NAME | DATE/TIME | ACCESSED SITE |
|---|---|---|---|
| 172.16.16.211 | 172.16.16.211 | 21/01/2014-12:05:04 | aus3.mozilla.org:443 |
| | | 21/01/2014-10:48:34 | fhr.data.mozilla.com:443 |
| | | 21/01/2014-11:04:30 | fhr.data.mozilla.com:443 |
| | | 21/01/2014-12:04:38 | fhr.data.mozilla.com:443 |
| | | 21/01/2014-12:11:25 | services.addons.mozilla.org:443 |
| | | 21/01/2014-12:11:25 | versioncheck-bg.addons.mozilla.org:443 |
| | | 21/01/2014-12:11:25 | versioncheck-bg.addons.mozilla.org:443 |
| | | 21/01/2014-12:11:25 | versioncheck-bg.addons.mozilla.org:443 |
| 172.16.21.234 | 172.16.21.234 | 21/01/2014-04:22:22 | http://si.informer.com |
| 172.16.24.230 | 172.16.24.230 | 21/01/2014-07:31:41 | http://www.msftncsi.com |
| 172.16.26.1 | 172.16.26.1 | 21/01/2014-12:36:39 | http://172.16.25.252 |
| 172.16.26.2 | 172.16.26.2 | 21/01/2014-12:30:10 | http://sa.windows.com |
| | | 21/01/2014-12:30:10 | http://sa.windows.com |
| | | 21/01/2014-12:30:13 | http://sa.windows.com |
| | | 21/01/2014-12:31:36 | http://sa.windows.com |
| | | 21/01/2014-12:31:58 | http://sa.windows.com |
| 172.16.26.3 | 172.16.26.3 | 21/01/2014-10:13:52 | addons.mozilla.org:443 |
| | | 21/01/2014-08:54:31 | aus3.mozilla.org:443 |
| | | 21/01/2014-08:48:35 | http://archive.mid-day.com |

**Fig 7. Denied Access**




| USERID | IP/NAME | DATE/TIME | ACCESSED SITE |
|---|---|---|---|
| 172.16.144.114 | 172.16.144.114 | 21/01/2014-12:21:26 | accounts.google.com:443 |
| | | 21/01/2014-12:21:26 | accounts.google.com:443 |
| | | 21/01/2014-12:21:26 | accounts.google.com:443 |
| | | 21/01/2014-12:21:26 | accounts.google.com:443 |
| | | 21/01/2014-12:21:26 | accounts.google.com:443 |
| | | 21/01/2014-12:21:27 | accounts.google.com:443 |
| | | 21/01/2014-12:21:27 | accounts.google.com:443 |
| | | 21/01/2014-12:21:28 | accounts.google.com:443 |
| | | 21/01/2014-12:21:30 | accounts.google.com:443 |
| | | 21/01/2014-12:21:30 | accounts.google.com:443 |
| | | | 77 more authentication failures not shown here... |
| 172.16.144.130 | 172.16.144.130 | 21/01/2014-09:24:09 | ent-shasta-rrs.symantec.com:443 |
| | | 21/01/2014-09:34:46 | ent-shasta-rrs.symantec.com:443 |
| | | 21/01/2014-09:45:09 | ent-shasta-rrs.symantec.com:443 |
| | | 21/01/2014-09:05:01 | http://172.16.16.70:8014 |
| | | 21/01/2014-09:47:23 | http://ad.goo.mx |
| | | 21/01/2014-09:04:59 | http://defender:8014 |
| | | 21/01/2014-09:05:01 | http://defender:8014 |
| | | 21/01/2014-09:05:00 | http://defender.midcorp.mid-day.com:8014 |

**Fig 8. Authentication Failures**

### 4.5 Assignment Questions:

1. Why to Configure Proxy Server?
2. What is SARG?
3. Which Parameter is there in SARG Report?
4. What do you mean by Log and Event Co-relation?

**Conclusion:**

By configuring this Network Administrator can easily analyze the Network Traffic and Bandwidth Utilization.

**Assignment Group A-5**

**Problem Definition:**

Design and Implement of Honeypot

**Problem statement:** Study and Implementation of Honeypot

**Learning objective:**

- To learn the concept of Honeypot
- To study the representation, implementation of Honeypot.

**Learning outcome:**

- Use honeybot tool to capture packets and configure tools and systems to enter unknown unauthenticated IP.

**Software and hardware requirement:**

- 64 bit machine
- Windows 7/8 Operating System

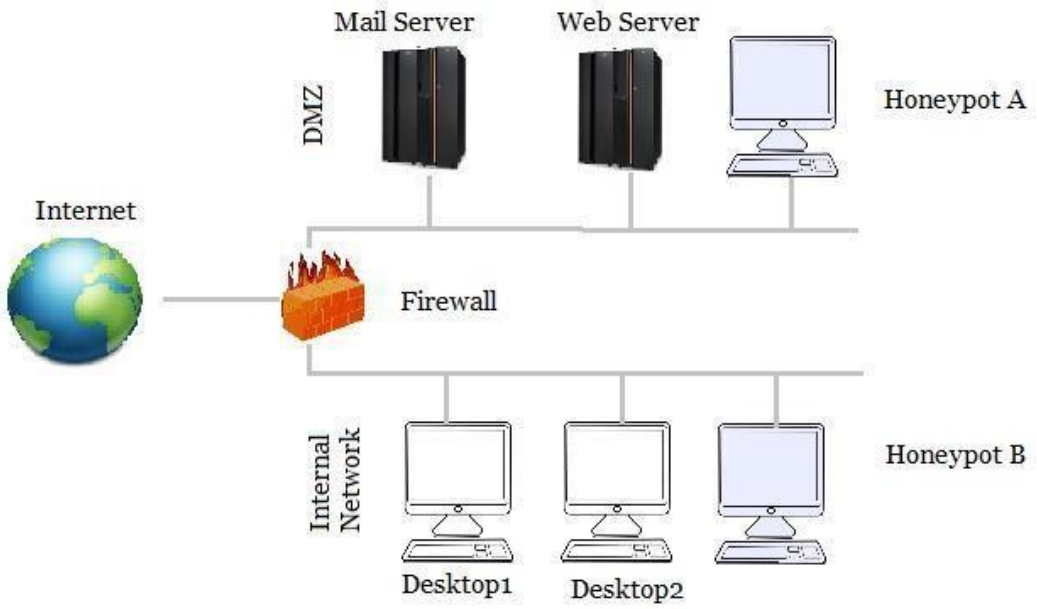
**5.1 Theory:**

**Honeypot:**

It is a computer system. There are files, directories in it just like a real computer. However, the aim of the computer is to attract hackers to fall into it to watch and follow their behavior. So we can define it as a fake system which looks like a real system. They are different than other security systems since they are not only finding one solution to a particular problem, but also they are eligible to apply variety of security problems and finding several approaches for them. For example, they can be used to log

Malicious activities in a compromised system; they can be also used to learn new threats for users and creating ideas how to get rid of those problems.

Honeypots are security resources that have no production value; no person or resource should be communicating with them. Any activity sent their way is suspect. Any traffic initiated by the honeypot means the system has most likely been compromised. Any traffic sent to the honeypot is most likely a probe, scan, or attack. With a honeypot, nothing is expected. To better understand the concepts of honeypots, let's take a look at the following example of honeypot deployments. Refer figure 11.1.




The purpose here is to demonstrate to you that honeypots can come in many different flavors, and they can achieve different things. However, they are both honeypots because they share the same definition and concepts. With the intent using systems as a honeypots, to determine if there is any unauthorized activity happening within your DMZ.

Honeypots passively capture any traffic or activity that interacts with them

### 5.1.1 Types of Honeypots:

There are two general types of honeypots:

5.1.1.1 **Production honeypots** are easy to use, capture only limited information, and are used primarily by companies or corporations. They are capturing a limited amount of information; mostly low interaction honeypots are used. security administrator watches the hacker's movements carefully and tries to lower the risks that may come from it towards the company

5.1.1.2 **Research honeypots** are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations. The objective is to learn how to Protect a system better, they do not bring any direct value to the security of an organization Honeypots are increasingly used to provide early warning of potential intruders, identify flaws in security strategies, and improve an organization's overall security awareness. "Honeypots can simulate a variety of internal and external devices, including Web servers, mail servers, database servers,

application servers, and even firewalls. As a software development manager, we can regularly use honeypots to gain insight into vulnerabilities in both the software my team writes and the OS upon which we depend."

A honeypot is a security resource whose value lies in being probed, attacked, or compromised. This means that whatever we designate as a honeypot, it is our expectation and goal to have the system probed, attacked, and potentially exploited.

## 5.2 Legal issues with honeypots

While deploying and start using a honeypot, there are some legal issues that a person should know about. Every country has different laws regarding to honeypot usage and information capturing. These regulations are related to data security, collection of data and finally how to use honeypots. All these different laws are based on the quality of the data that a honeypot can capture and a person who is deploying it. Privacy and data leads us to confidentiality term in network security. Our example is being a network administrator in a company.

## 5.3 Practical implementation

We are starting with low interaction honeypot and then continue on a middle level of interaction to finally conclude with a high level of interaction.

- **Starting to honeypot**,

We started with Honeyd as low level interaction honeypot and then we will move on medium level interaction honeypots. Every honeypot has specific and different attitudes. We will explain them one by one.

- **HoneyBOT is a medium interaction honeypot for windows.**

A honeypot creates a safe environment to capture and interact with unsolicited and often malicious traffic on a network. HoneyBOT is an easy to use solution ideal for network security research or as part of an early warning IDS. The logging capability of a honeypot is far greater than any other network security tool and captures raw packet level data even including the keystrokes and mistakes made by hackers. The captured information is highly valuable as it contains only malicious traffic with little to no false positives. Honeypots are becoming one of the leading security tools used to monitor the latest tricks and exploits of hackers by recording their every move so that the security community can more quickly respond to new exploits.

- **How does it work?**

HoneyBOT works by opening a range of listening sockets on your computer which are designed to mimic vulnerable services. When an attacker connects to these services they are fooled into thinking they are attacking a real server. The honeypot safely captures all communications with the attacker and logs these results for future analysis. Should an attacker attempt an exploit or upload a rootkit or trojan to the server the honeypot environment can safely store these files on your computer for malware collection and analysis purposes. Following figure shows implementation of honeypot.

- **Installing and Securing Your Honeypot**

A honeypot is intentionally put in harms way so it is critical to carry out some security precautions on your honeypot computer before deployment on any network. Install HoneyBOT on a dedicated computer or virtual machine. Update the operating system with security updates and use an antivirus product. You want your honeypot to be as free as possible from legitimate traffic so in broad terms we can consider any traffic to the honeypot to be malicious in nature. Remember that we are attracting attackers to intrude into this system so precautions are important.

- **Network Placement**

If you place HoneyBOT inside the internal network where it is secured by perimeter defences it should never to be attacked. Any malicious traffic captured in this situation would indicate that another computer inside the network is already compromised or that the perimeter defences have been breached. In this configuration HoneyBOT is acting as an intrusion detection system. If you place HoneyBOT on an external network or internet you will attract higher volumes of unsolicited network traffic. Direct internet placement is the most common setup with HoneyBOT being on the network DMZ.

- **Windows Services, SMB and NetBIOS**

You should disable any Windows services that are not required for the machine to operate as they offer an attacker a possible avenue of attack. HoneyBOT cannot listen on a port that is already in use by a Windows service. Some of the services that you may choose to disable include Messenger, ClipBook, COM+, FTP Publishing, SMTP, SNMP, TCP/IP NetBIOS Helper, Telnet, WWW Publishing.

SMB (CIFS) provides name resolution, network browsing and printing services over TCP/IP. To disable SMB open the Network Connections window, right click the adapter and select Properties and uninstall Client For Microsoft Networks and File And Printer Sharing. SMB services may also be provided over NetBIOS (NBT). To disable NetBIOS open the Device Manager window, select Show Hidden Devices, expand Non-Plug And Play Drivers and disable NetBios Over Tcpip.

If you are monitoring your honeypot via a remote desktop tool then you should change the default listening port to a random high numbered port.

Finally, before starting HoneyBOT take a baseline of the current listening services by opening a command shell and launching netstat with the -ano option. Any listening services that you are unable to disable need to be blocked at the firewall.

- **Firewall**

A firewall will prevent unsolicited connections from reaching your computer. In order for HoneyBOT to communicate you need to customise your firewall rules to allow incoming connections. If you are using a software firewall you should create an exception for HoneyBOT.

- **HoneyBOT Options**

Select Options from the View menu to configure HoneyBOT.

Automatically Start Engine: The server engine will start automatically when the application is started.

Enable Sound Alert: Plays a short sound each time an event occurs. Capture Binaries: If this option is enabled HoneyBOT will attempt to capture malware and other files and save them to the \HoneyBOT\Captures\ folder. If this option is enabled you should add an exception in your antivirus software to exclude this folder from its scan. Automatically Rotate

Log: Each day at midnight HoneyBOT will save the current log file and start a new log file. Server Name: The alias name of the HoneyBOT server given to the remote machine.

- **Email Alerts**

Enter your email address and SMTP server information to receive daily email updates from HoneyBOT.

- **Exports**

Select the Export Logs to CSV option to create a daily extract of your log file as a CSV file. Exported logs are saved in the \HoneyBOT\Logs\ folder. You can also choose to participate in the centralised log program and have your log files uploaded to the HoneyBOT website.

- **Syslog**

Select to send connection events to a Syslog server. Enter the Syslog server IP address and

port.

- **Bindings**

Only applicable to multihomed machines. Provides support for multiple networks so HoneyBOT can bind to one or all detected networks. Enter the IP address that you want HoneyBOT to bind to. If the IP address is not valid and more than one IP address is available you will be prompted to select an address when the server engine starts.

- **Updates**

Select to have HoneyBOT check for updates on startup. There are two update types that may occur. A service update is a minor update to the server listening services, if a service update is available you will be prompted to install the update. An application update notification will occur if a new version of HoneyBOT is available.

- **Services and Profiles**

Select to edit the TCP and UDP services started by the HoneyBOT engine. You can add a new port, edit and disable an existing port, or delete the port configuration entirely.
By default HoneyBOT will open more listening ports than a typical computer and this may alert an attacker to its presence. You can choose to limit your honeypot exposure to just a handful of ports that more closely resembles a real operating system. By loading a profile you can quickly emulate common operation system setups like an SQL Server, IIS Server, Exchange Server, etc.

- **Whitelist**

You may find HoneyBOT is interacting with services on your network that are legitimate and not a cause for alarm. You can whitelist the source machine by adding the IP and port to the whitelist settings. When a machine is whitelisted HoneyBOT will no longer accept connections from that machine.

- **Debug**

The debug window will display application messages and socket events that occur during typical application operation.

- **Event Navigation**

The event tree on the left shows the ports that have been probed and remote addresses

that have connected to HoneyBOT. The event list at the top right will display all connection attempts including the attributes of the connection. The packet list at the bottom displays each packet transmitted and received between the remote machine and the HoneyBOT server. You can expand the event tree and filter the events displayed by selecting an item in the list.

### 5.4 Advantages of honeypots

There are many security solutions available in the market. Anyone can browse the variety of choices through internet and find the most suitable solution for their needs. Honeypots can capture attacks and give information about the attack type and if needed, thanks to the logs, it is possible to see additional information about the attack. New attacks can be seen and new security solutions can be created by looking at them. More examinations can be obtained by looking at the type of the malicious behaviors. It helps to understand more attacks that may happen. Honeypots are not bulky in terms of capturing data. They are only dealing with the incoming malicious traffic. Therefore, the information that has been caught is not as much as the whole traffic. Focusing only on the malicious traffic makes the investigation far easier.

### 5.5 Disadvantages of honey pots

We can only capture data when the hacker is attacking the system actively. If he does not attack the system, it is not possible to catch information. If there is an attack occuring in another system, our honeypot will not be able to identify it. So, attacks not towards our honeypot system may damage other systems and cause big problems. There is fingerprinting disadvantage of honeypots. It is easy for an experienced hacker to understand if he is attacking a honeypot system or a real system. Fingerprinting allows us to distinguish between these two. It is a not a wanted result of our experiment. The honeypot may be used as a zombie to reach other systems and compromise them. This can be very dangerous.

## 5.6 Assignment:

Q1.What is Honey Pot?

Q2.What are different types of Honey Pot?

Q3.What is Malware Honey Pot?

Q4. What is Database honey pot?
Q5.What is Honey nets?
Q6. Which are two popular reasons or goals behind setting up a Honey Pot?

**Conclusion:**

Hence, we have successfully studied concept of Honeypot in which we have set different network setting and set different drivers to identify unauthenticated access in our system

**GROUP-2**

1. Mini–project: Perform the following steps:

   - Go to the National Child Exploitation Coordination Centre (NCECC) Web site at http:// www.ncecc.ca
   - Click on the Reporting child exploitation link.
   - c. Read "How to Report Internet Pornography or Internet Luring Related to Children."

   OR

2. Mini- Project: Perform the following steps:

   - Go to http://www.usdoj.gov/criminal/cybercrime/cyberstalking.htm.
   - b. Read the 1999 report on cyber stalking.

PRESS RELEASE

# Member of Violent Extremist Network '764' Charged with Animal Crushing, Sexual Exploitation of a Minor, Cyberstalking and Interstate Threats

Monday, October 27, 2025

**For Immediate Release**

Office of Public Affairs

A federal grand jury in the Eastern District of California has returned a six-count indictment against Tony Christopher Long, also known as Inactive, Inactivee0, and inactivecvx, 19, of Porterville, California, charging him with animal crushing (two counts), sexual exploitation of a minor, possession of material involving the sexual exploitation of a minor, cyberstalking, and transmitting an interstate threat. Long is currently in state custody on related charges.

"This defendant allegedly engaged in acts of extreme cruelty by exploiting a child, abusing animals, and threatening violence — his conduct reflects the depravity of '764,'" said Attorney General Pamela Bondi. "These networks seek to terrorize and destabilize our communities by preying on the most vulnerable, and the Justice Department will stop at nothing to dismantle this network and bring offenders to justice."

"The FBI has no tolerance for anyone who preys on children or other vulnerable members of society," said FBI Director Kash Patel. "This defendant allegedly targeted juveniles, took part in animal crushing, and was part of a violent online network which seeks to sow chaos and destabilize our society. The FBI will work with our law enforcement partners to investigate and hold accountable anyone who engages in such reprehensible and illegal activity."

"This indictment charges a constellation of offenses related to the troubling emergence of NVEs like '764' and related groups," said U.S. Attorney Eric Grant for the Eastern District of

California. "My office will vigorously investigate and prosecute offenses committed by NVE groups, including those alleged to have been committed by Long against young and vulnerable victims," he added.

According to court documents, Long was a member and associate of "764," a criminal organization of Nihilistic Violent Extremists (NVEs). NVEs are individuals who engage in criminal conduct within the United States and abroad in furtherance of political, social, or religious goals that derive primarily from a hatred of society and a desire to bring about its collapse via chaos, destruction, and social instability. NVEs work individually or as part of a network with the goal of destroying civilized society through the corruption and exploitation of vulnerable populations, which often include minors.

The indictment, returned by the grand jury on Oct. 23, alleges that in late 2024, Long purposely engaged in animal crushing, sexually exploited a juvenile victim living in Washington state, and committed cyberstalking and made online threats against a juvenile victim living in Kern County, California.

If convicted, Long faces a maximum penalty of seven years in prison on each of the two counts charging animal crushing; a minimum mandatory penalty of 15 years in prison up to a maximum of 30 years in prison for sexual exploitation of a minor; a maximum penalty of 10 years in prison for possession of material involving the sexual exploitation of a minor; a maximum penalty of 20 years in prison for cyberstalking; and a maximum statutory penalty of two years in prison for making an interstate threat. Each count of the indictment also carries a fine of up to $250,000. A federal district court judge will determine any sentence after considering the U.S. Sentencing Guidelines and other statutory factors.

The FBI is investigating the case, with assistance from the Porterville Police Department.

This case was brought as part of Project Safe Childhood, a nationwide initiative to combat the epidemic of child sexual exploitation and abuse launched in May 2006 by the Department of Justice. Led by United States Attorneys' Offices and the Criminal Division's Child Exploitation and Obscenity Section (CEOS), Project Safe Childhood marshals federal, state, and local resources to better locate, apprehend and prosecute individuals who exploit children via the Internet, as well as to identify and rescue victims. For more information about Project Safe Childhood, please visit www.justice.gov/psc.

The Justice Department remains vigilant against the threat of Nihilistic Violent Extremist (NVE) networks, like 764, that operate within the United States and around the globe. NVEs often target vulnerable individuals, including minors, using social media platforms to share child sexual abuse material (CSAM) and gore material, and groom victims toward committing acts of violence. Victims are often extorted, coerced, compelled, and blackmailed into complying with NVE demands, including self-mutilation, online and in-person sexual acts, harm to animals, sexual exploitation of siblings and others, acts of violence, threats of violence, suicide, and murder. For more information on how to protect children and others,

read about the online risks here: Parents, Caregivers, Teachers — FBI and the FBI's March 2025 public service announcement.

*An indictment is merely an allegation. All defendants are presumed innocent until proven guilty beyond a reasonable doubt in a court of law.*

*Updated October 27, 2025*

**Components**

Office of the Attorney General | National Security Division (NSD) | USAO - California, Eastern

Press Release Number: 25-1043

# Related Content

PRESS RELEASE

## Former General Manager for U.S. Defense Contractor Pleads Guilty to Selling Stolen Trade Secrets to Russian Broker

Peter Williams, 39, an Australian national, pleaded guilty in U.S. District Court today in connection with selling his employer's trade secrets to a Russian cyber-tools broker, the Justice Department announced...

October 29, 2025

PRESS RELEASE

## Father and Son Arrested for Attempting to Smuggle Hundreds of Firearms to Mexico

Two men from Alabama have been charged with trafficking more than 300 weapons along with ammunition and magazines, announced Attorney General Pamela Bondi and U.S.

Attorney Nicholas J. Ganjei.

October 28, 2025

VIDEO

## Third Hearing of the Religious Liberty Commission, Part 1

September 29, 2025

**Office of Public Affairs**

U.S. Department of Justice

950 Pennsylvania Avenue, NW

Washington DC 20530

Office of Public Affairs Direct Line

202-514-2007

Department of Justice Main Switchboard

202-514-2000

DHOLE PATIL EDUCATION SOCIETY's
DHOLE PATIL COLLEGE OF ENGINEERING
**Accredited by NAAC with A+ Grade, An ISO 9001:2015 Certified Institute**
**1284, Near Eon IT Park Kharadi, Dhole Patil College Road, Wagholi, Pune-412207**
**Website: www.dpcoepune.edu.in E-mail:** dpcoepune@gmail.com, Phone:020-6605990

## Department of Computer Engineering

## AY-2025-2026

# Laboratory Practice IV (410247)
# Lab Manual (STQA)

**Prepared By**

**Prof. Payal Nikam**

DHOLE PATIL EDUCATION SOCIETY's
DHOLE PATIL COLLEGE OF ENGINEERING
**Accredited by NAAC with A+ Grade, An ISO 9001:2015 Certified Institute**
**1284, Near Eon IT Park Kharadi, Dhole Patil College Road, Wagholi, Pune-412207**
**Website: www.dpcoepune.edu.in E-mail:** dpcoepune@gmail.com, Phone:020-6605990

# CERTIFICATE

This is to certify that, Mr./ Miss._____

Roll No:_____of **First Semester** of Fourth Year in **Computer Engineering (2019 Course)** has completed the term work satisfactorily in **Laboratory Practice IV (Elective III - 410244(C): Cyber Security and Digital Forensics and Elective IV 410245 (D): Software Testing and Quality Assurance)** for the Academic Year 2025-2026 (Sem-I) as prescribed in the curriculum of Savitribai Phule Pune University, Pune.

Place: PRN No.:

Date: Exam Seat No:

**Subject Teacher** **Head of Department** **Principal**

# Laboratory Practice IV Lab Manual (STQA) (2025-2026)

## Index


| Sr No | Assignment | Page No |
|---|---|---|
| 1 | Write TEST Scenario for Gmail Login Page | 1 |
| 2 | Test Scenario for Gmail – Inbox | 7 |
| 3 | Write Test cases in excel sheet for Social Media application or website | 9 |
| 4 | Create Defect Report for Any application or web application | 17 |
| 5 | Installation of Selenium grid and selenium Webdriver & java eclips (automation tools). | 28 |

# Assignment No.: 1

**AIM: -** Write TEST Scenario for Gmail Login Page

## THEORY:

## Software Testing:

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

**Software testing can be divided into two steps:**
1. **Verification:** it refers to the set of tasks that ensure that the software correctly implements a specific function.
2. **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

## Test Case:

The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios.

## Test Cases – Login Page

## Functional Test Cases

| Sr. No. | Functional Test Cases | Type- Negative/ Positive Test Case |
|---|---|---|
| 1 | Verify if a user will be able to login with a valid username and valid password. | Positive |
| 2 | Verify if a user cannot login with a valid username and an invalid password. | Negative |
| 3 | Verify the login page for both, when the field is blank and Submit button is clicked. | Negative |
| 4 | Verify the ‘Forgot Password’ functionality. | Positive |
| 5 | Verify the messages for invalid login. | Positive |
| 6 | Verify the ‘Remember Me’ functionality. | Positive |
| 7 | Verify if the data in password field is either visible as asterisk or bullet signs. | Positive |
| 8 | Verify if a user is able to login with a new password only after he/she has changed the password. | Positive |
| 9 | Verify if the login page allows to log in simultaneously with different credentials in a different browser. | Positive |
| 10 | Verify if the ‘Enter’ key of the keyboard is working correctly on the login page. | Positive |
| | Other Test Cases | |
| 11 | Verify the time taken to log in with a valid username and password. | Performance & Positive Testing |
| 12 | Verify if the font, text color, and color coding of the Login page is as per the standard. | UI Testing & Positive Testing |
| 13 | Verify if there is a ‘Cancel’ button available to erase the entered text. | Usability Testing |
| 14 | Verify the login page and all its controls in different browsers | Browser Compatibility & Positive Testing. |

## Non Functional Test Cases

| Sr. No. | Functional Test Cases | Type- Negative/ Positive Test Case |
|---|---|---|
| 1 | Verify if a user will be able to login with a valid username and valid password. | Positive |
| 2 | Verify if a user cannot login with a valid username and an invalid password. | Negative |
| 3 | Verify the login page for both, when the field is blank and Submit button is clicked. | Negative |
| 4 | Verify the 'Forgot Password' functionality. | Positive |
| 5 | Verify the messages for invalid login. | Positive |
| 6 | Verify the 'Remember Me' functionality. | Positive |
| 7 | Verify if the data in password field is either visible as asterisk or bullet signs. | Positive |
| 8 | Verify if a user is able to login with a new password only after he/she has changed the password. | Positive |
| 9 | Verify if the login page allows to log in simultaneously with different credentials in a different browser. | Positive |
| 10 | Verify if the 'Enter' key of the keyboard is working correctly on the login page. | Positive |
| | Other Test Cases | |
| 11 | Verify the time taken to log in with a valid username and password. | Performance & Positive Testing |
| 12 | Verify if the font, text color, and color coding of the Login page is as per the standard. | UI Testing & Positive Testing |
| 13 | Verify if there is a 'Cancel' button available to erase the entered text. | Usability |

| Sr. No. | Functional Test Cases | Type- Negative/ Positive Test Case |
|---|---|---|
| | | Testing |
| 14 | Verify the login page and all its controls in different browsers | Browser Compatibility & Positive Testing. |

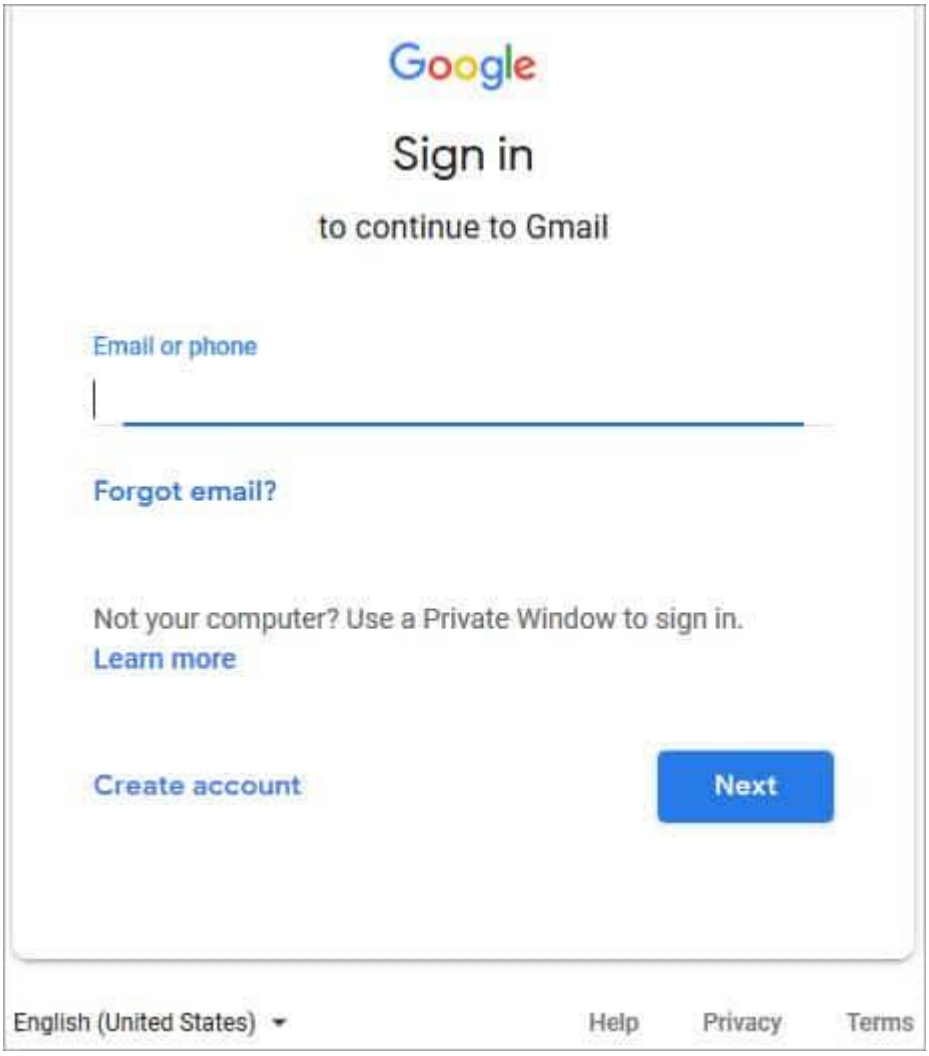



Fig. 1 Gmail Login page

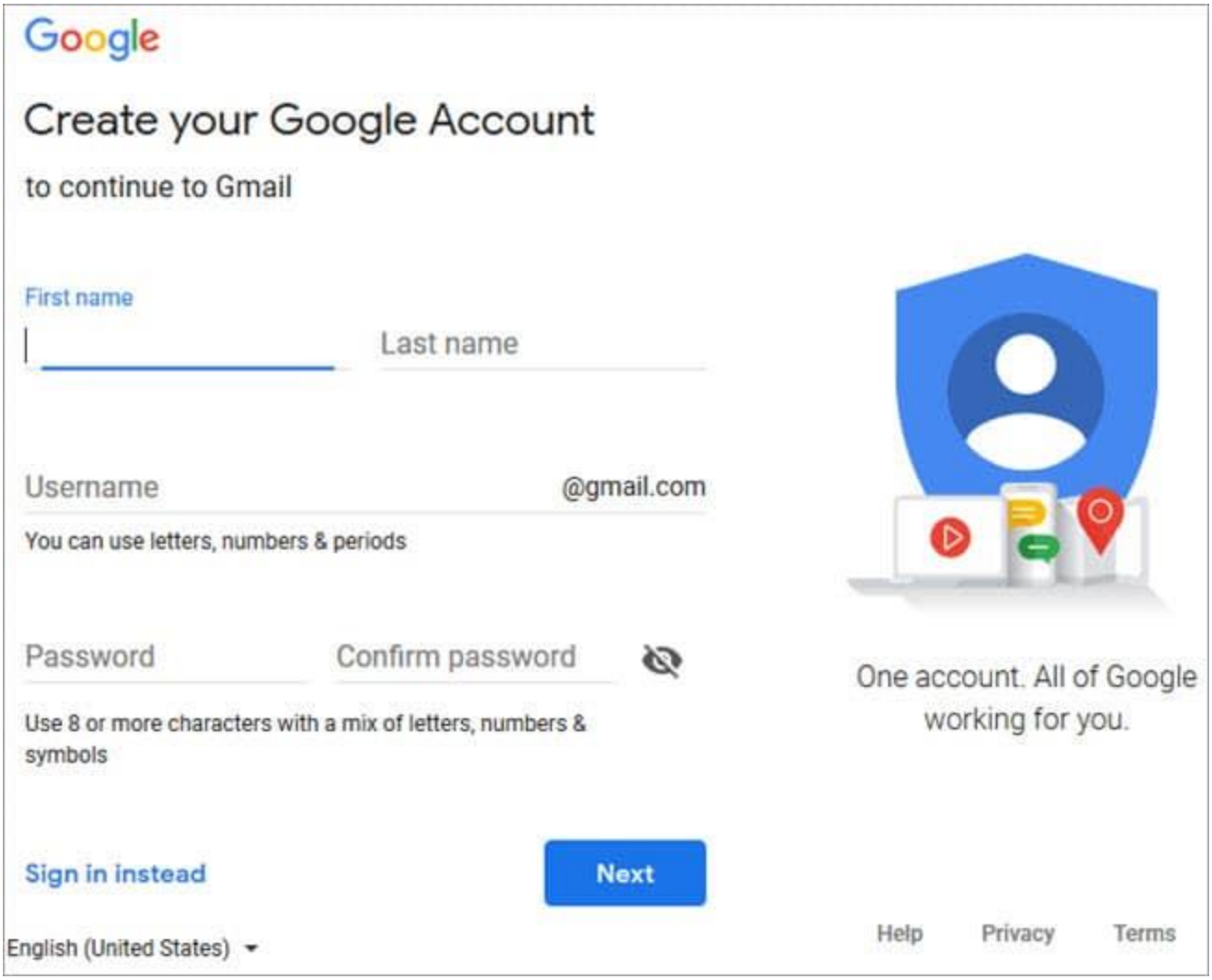



**Fig. 2 Test Case for Gmail Login Page**

## Test Scenarios for Gmail Login Page

| Sr. No. | Test Scenarios |
|---|---|
| 1 | Enter the valid email address & click next. Verify if the user gets an option to enter the password. |
| 2 | Don't enter an email address or phone number & just click the Next button. Verify if the user will get the correct message or if the blank field will get highlighted. |
| 3 | Enter the invalid email address & click the Next button. Verify if the user will get the correct message. |

| Sr. No. | Test Scenarios |
|---|---|
| 4 | Enter an invalid phone number & click the Next button. Verify if the user will get the correct message. |
| 5 | Verify if a user can log in with a valid email address and password. |
| 6 | Verify if a user can log in with a valid phone number and password. |
| 7 | Verify if a user cannot log in with a valid phone number and an invalid password. |
| 8 | Verify if a user cannot log in with a valid email address and a wrong password. |
| 9 | Verify the ‘Forgot email’ functionality. |
| 10 | Verify the ‘Forgot password’ functionality. |

## Conclusion:

Successfully Written TEST Scenario for Gmail Login Page

# Assignment No.:2

**AIM:-** Test Scenario for Gmail – Inbox Functionality

1) Verify that a newly received email is displayed as highlighted in the Inbox section.

2) Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body(trimmed to a single line).

3) Verify that on clicking the newly received email, the user is navigated to email content.

4) Verify that the email contents are correctly displayed with the desired source formatting.

5) Verify that any attachments are attached to the email and are downloadable.

6) Verify that the attachments are scanned for viruses before download.

7) Verify that all the emails marked as read are not highlighted.

8) Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.

9) Verify that count of unread emails is displayed alongside 'Inbox' text in the left sidebar of Gmail.

10) Verify that unread email count increases by one on receiving a new email.

11) Verify that unread email count decreases by one on reading an email ( marking an email as read).

12) Verify that email recipients in cc are visible to all users.

13) Verify that email recipients in bcc are not visible to the user.

14) Verify that all received emails get piled up in the 'Inbox' section and get deleted in cyclic fashion based on the size availability.

15) Verify that email can be received from non-Gmail email Ids like – yahoo, Hotmail etc.

Test Cases for GMail – Compose Mail Functionality

1) Verify that on clicking 'Compose' button, a frame to compose a mail gets displayed.

2) Verify that user can enter email Ids in 'To', 'cc' and 'bcc' sections and also user will get suggestions while typing the emailds based on the existing emailIds in user's email list.

3) Verify that the user can enter multiple comma-separated emailIds in 'To', 'cc' and 'bcc' sections.

4) Verify that the user can type Subject line in the 'Subject' textbox.

5) Verify that the user can type the email in the email-body section.

6) Verify that users can format mail using editor-options provided like choosing font-family, font-size, bold-italic-underline, etc.

7) Verify that the user can attach file as an attachement to the email.

8) Verify that the user can add images in the email and select the size for the same.

9) Verify that after entering emailIds in either of the 'To', 'cc' and 'bcc' sections, entering Subject line and mail body and clicking 'Send' button, mail gets delivered to intended receivers.

10) Verify that sent mails can be found in 'Sent Mail' sections of the sender.

11) Verify that mail can be sent to non-gmail emailIds also.

12) Verify that all sent emails get piled up in the 'Sent Mail' section and get deleted in cyclic fashion based on the size availability.

13) Verify that the emails composed but not sent remain in the draft section.

14) Verify the maximum number of email recipients that can be entered in 'To', 'cc' and 'bcc' sections.

15) Verify the maximum length of text that can be entered in the 'Subject' textbox.

16) Verify the content limit of text/images that can be entered and successfully delivered as mail body.

17) Verify the maximum size and number of attachment that can be attached with an email.

18) Verify that only the allowed specifications of the attachment can be attached with an email/

19) Verify that if the email is sent without Subject, a pop-up is generated warning user about no subject line. Also,

20) Verify that on accepting the pop-up message, the user is able to send the email.

**Conclusion**: Successfully studied Test Scenario for Gmail – Inbox Functionality.

# Assignment No.:3

**AIM:-** Write Test cases in excel sheet for Social Media application or website

## Theory:

## Test Case:

The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios.

Sample Test Case Template With Test Case Examples

If you are not using any Test case management tool, then I would strongly recommend you to use an open- source tool to manage and execute your test cases.

Test case formats may vary from one organization to another. However, using a standard test case format forwriting test cases is one step closer to setting up a testing process for your project.

It also minimizes Ad-hoc testing that is done without proper test case documentation. But even if you use standard templates, you need to set up test cases writing, review & approve, test execution and most importantly test report preparation process, etc. by using manual methods.

Also, if you have a process to review the test cases by the business team, then you must format these test casesin a template that is agreed by both the parties.

Recommended Tools

Before continuing with the Test case writing process, we recommend downloading these Test case management tools. This will ease your test plan and test case writing process mentioned in this tutorial.

1) TestRail

TestRail is a web-based tool for test cases and test management. It helps QA and development teams with the efficient management of test cases, plans, and runs. It gives centralized test management, powerful reports & metrics, and increased

productivity. It is a scalable and customizable solution. It can be used by small as well aslarge teams.

Features:

1) TestRail makes tracking test results easier.
2) It seamlessly gets integrated with bug trackers, automated tests, etc.
3) Personalized to-do lists, filters, and email notifications will help with boosting productivity.
4) Dashboards and activity reports are for easy tracking and following the status of individual tests,milestones, and projects.

2) Katalon Studio

Katalon Studio is an all-in-one, simple automation tool for web, API, mobile, and desktop trusted by over850,000 users.

It simplifies automation for those without a coding background to create automation test cases from manualtests' steps, a rich library of project templates, record & playback, and a friendly UI.

3) Testiny

Testiny – a new, straightforward test management tool, but much more than just a slimmed-down app. Testiny is a fast-growing web application built on the latest technologies and aims to make manual testing andQA management as seamless as possible. It is designed to be extremely easy to use. It helps testers perform tests without adding bulky overhead to the testing process.

Don't just take our word for it, take a look at Testiny yourself. Testiny is perfect for small to mid-sized QA teams looking to integrate manual and automated testing into their development process.

Features:

1) Free for open-source projects and small teams with up to 3 people.
2) Intuitive and simple out of the box.
3) Easily create and handle your test cases, test runs, etc.
4) Powerful integrations (e.g. Jira, …)
5) Seamless integration in the development process (linking requirements and defects)
6) Instant updates – all browser sessions stay in sync.
7) Immediately see if a colleague has made changes, completed a test, etc.
8) Powerful REST API.

9) Organize your tests in a tree structure – intuitive and easy.

Standard Fields of a Sample Test Case Template

There are certain standard fields that need to be considered while preparing a Test case template.

| | |
|---|---|
| **Project Name:** | |
| **Test Case Template** | |
| **Test Case ID:** Fun_10 | **Test Designed by:** <Name> |
| **Test Priority (Low/Medium/High):** Med | **Test Designed date:** <Date> |
| **Module Name:** Google login screen | **Test Executed by:** <Name> |
| **Test Title:** Verify login with valid username and password | **Test Execution date:** <Date> |
| **Description:** Test the Google login page | |
| | |
| | |
| **Pre-conditions:** User has valid username and password | |
| **Dependencies:** | |

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|---|---|---|---|---|---|---|
| 1 | Navigate to login page | User= example@gmail.com | User should be able to login | User is navigated to | Pass | |
| 2 | Provide valid username | Password: 1234 | | dashboard with successful | | |
| 3 | Provide valid password | | | login | | |
| 4 | Click on Login button | | | | | |
| | | | | | | |

**Post-conditions:**

User is validated with database and successfully login to account. The account session details are logged in database.

Several standard fields for a sample Test Case template are listed below.

1) Test case ID: Unique ID is required for each test case. Follow some conventions to indicate the types of the test. For Example, 'TC_UI_1' indicating 'user interface test case #1'.
2) Test priority (Low/Medium/High): This is very useful during test execution. Test priorities for businessrules and functional test cases can be medium or higher, whereas minor user interface cases can be of a low priority. Testing priorities should always be set by the reviewer.
3) Module Name: Mention the name of the main module or the sub-module.
4) Test Designed By: Name of the Tester.
5) Test Designed Date: Date when it was written.
6) Test Executed By Name of the Tester who executed this test. To be filled only after

test execution.

7) Test Execution Date: Date when the test was executed.
8) Test Title/Name: Test case title. For example, verify the login page with a valid username and password.
9) Test Summary/Description: Describe the test objective in brief.
10) Pre-conditions: Any prerequisite that must be fulfilled before the execution of this test case. List all thepre-conditions in order to execute this test case successfully.
11) Dependencies: Mention any dependencies on other test cases or test requirements.
12) Test Steps: List all the test execution steps in detail. Write test steps in the order in which they should beexecuted. Make sure to provide as many details as you can.
13) Test Data: Use of test data as an input for this test case. You can provide different data sets with exactvalues to be used as an input.
14) Expected Result: What should be the system output after test execution? Describe the expected result indetail including the message/error that should be displayed on the screen.
15) Post-condition: What should be the state of the system after executing this test case?
16) Actual result: The actual test result should be filled after test execution. Describe the system behavioafter test execution.
17) Status (Pass/Fail): If the actual result is not as per the expected result, then mark this test as failed.Otherwise, update it as passed.
18) Notes/Comments/Questions: If there are any special conditions to support the above fields, which can'tbe described above or if there are any questions related to expected or actual results then mention themhere.

Add the following fields if necessary:

1) Defect ID/Link: If the test status fails, then include the link to the defect log or mention the defectnumber.
2) Test Type/Keywords: This field can be used to classify tests based on test types. For Example,functional, usability, business rules, etc.

3) Requirements: Requirements for which this test case is being written for. Preferably the exact sectionnumber in the requirement doc.
4) Attachments/References: This field is useful for complex test scenarios in order to explain the test stepsor expected results using a Visio diagram as a reference. Provide a link or location to the actual path of the diagram or document.
5) Automation? (Yes/No): Whether this test case is automated or not. It is useful to track automation statuswhen test cases are automated.

One More Test Case Format (#2)

The test cases will differ depending upon the functionality of the software that it is intended for. However, given below is a template that you can always use to document the test cases without bothering about what yourapplication is doing?

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Test Scenario ID** | | | **Test Case ID** | | | | |
| **Test Case Description** | | | **Test Priority** | | | | |
| **Pre-Requisite** | | | **Post-Requisite** | | | | |
| Test Execution Steps: | | | | | | | |
| **S.No** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Browser** | **Test Result** | **Test Comments** |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Sample Test Cases

Based on the above template, below is an example that showcases the concept in a much understandable way. Let's assume that you are testing the login functionality of any web application, say Facebook.

**An ideal test case template**

Below is a template which you can always use for documenting the test cases without bothering with what yourapplication is doing.

| Test Scenario ID | | Test Case ID | |
|---|---|---|---|
| Test Case Description | | Test Priority | |
| Pre-Requisite | | Post-Requisite | |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result | Test Comments |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

**Example Scenario**

Based on the above template, below is an example that showcases the concepts in a more understandable way.

Suppose you are testing the login functionality of any web application, say Facebook. Below are the test casesfor the same:

| Test Scenario ID | Login-1 | Test Case ID | Login-1A |
|---|---|---|---|
| Test Case Description | Login – Positive test case | Test Priority | High |
| Pre-Requisite | A valid user account | Post-Requisite | NA |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result | Test Comments |
|---|---|---|---|---|---|---|---|
| 1 | Launch application | https://www.facebook.com/ | Facebook home | Facebook home | IE -11 | Pass | [Priya 12/17/2016 11:44 AM]: Launch successful |
| 2 | Enter correct Email & Password and hit login button | Email id : test@xyz.com Password: ****** | Login success | Login success | IE -11 | Pass | [Priya 12/17/2016 11:45 AM]: Login successful |
| | | | | | | | |

| | | | |
|---|---|---|---|
| **Test Scenario ID** | Login-1 | **Test Case ID** | Login-1B |
| **Test Case Description** | Login – Negative test case | **Test Priority** | High |
| **Pre-Requisite** | NA | **Post-Requisite** | NA |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result | Test Comments |
|---|---|---|---|---|---|---|---|
| 1 | Launch application | https://www.facebook.com/ | Facebook home | Facebook home | IE -11 | Pass | [Priya 12/17/2016 11:44 AM]: Launch successful |
| 2 | Enter invalid Email & any Password and hit login button | Email id : invalid@xyz.com Password: ****** | The email address or phone number that you've entered doesn't match any account. Sign up for an account. | The email address or phone number that you've entered doesn't match any account. Sign up for an account. | IE -11 | Pass | [Priya 12/17/2016 11:45 AM]: Invalid login attempt stopped |
| 3 | Enter valid Email & incorrect Password and hit login button | Email id : valid@xyz.com Password: ****** | The password that you've entered is incorrect. Forgotten password ? | The password that you've entered is incorrect. Forgotten password? | IE -11 | Pass | [Priya 12/17/2016 11:46 AM]: Invalid login attempt stopped |

**Samples from a Live Project**

Below is examples from a live project that demonstrates how all the above listed tips and

tricks are actuallyimplemented:

| Timestamp | Description | Which Page? | Issue Type | Screenshot | Urgency | Browser | Re-test Date | Testing Comments |
|---|---|---|---|---|---|---|---|---|
| 2/7/13 17:08 | Clicking on enter after entering data closes the modal<br>Steps:<br>a) Click on 'Create New Patient'<br>b) Enter patient name and click enter<br>c) Create patient modal is closed | Login | Error or Bug | None | Medium | Firefox | 2/21/2013 | Pending |

Summary | Bugs | IMS-Search | Interaction | Quick Actions | HRA | Personal Health History ...

| Test Scenario Group | Test Case Id | Test Case Description | Test Input | Expected Result | Actual Result | Test Browser | Executed Date | Test Results |
|---|---|---|---|---|---|---|---|---|
| **Health records** | | | | | | | | |
| **Personal Health History** | PH 1 | High Cholestrol: | a) Yes and on medication | how old were you at time of dx? | how old were you at time of dx? | Chrome,IE, Firefox | | |
| | | | b) Yes but not on medication | how old were you at time of dx? | how old were you at time of dx? | Chrome,IE, Firefox | | |
| | PH 2 | High Blood Pressure: | a) Yes and on medication | what does your blood pressure range, on average? | what does your blood pressure range, on average? | Chrome,IE, Firefox | | |
| | | | b) Yes but not on medication | what does your blood pressure range, on average? | what does your blood pressure range, on average? | Chrome,IE, Firefox | | |
| | | | a) Yes and on medication | how old were you at time of dx? | how old were you at time of dx? | Chrome,IE, Firefox | | |
| | | | b) Yes but not on medication | how old were you at time of dx? | how old were you at time of dx? | Chrome,IE, Firefox | | |
| | PH 3 | Diabetes: | a) Yes and on medication | how old were you at time of dx? | | Chrome,Firefox | | |
| | | | b) Yes but not on medication | how old were you at time of dx? | | Chrome,Firefox | | |
| | | | a) Yes and on medication | what type of dm do you have? | | Chrome,Firefox | | |

| Test Scenario Group | Test Case Id | Test Case Description | Test Env | Test Input | Expected Result | Actual Result | Test Browser | Executed Date | Test Results | Executed Date | Test Results | Defect Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Search | Search-1 | Search for Patient only in Copia | | | | | | | | | | |
| Search | Search-2 | Search for Patient only in IMS | | | | | | | | | | |
| Search | Search-3 | Search for Patient in both Copia & IMS | | | | | | | | | | |
| Search | Search-4 | Search by first name | | | | | | | | | | |
| Search | Search-5 | Search by last name | | | | | | | | | | |
| Search | Search-6 | Search by patient id | | | | | | | | | | |
| Search | Search-7 | Search with partial name | | | | | | | | | | |
| Search | Search-8 | Search with partial patient id | | | | | | | | | | |
| Search | Search-9 | Update Patient profile | | | | | | | | | | |
| | | a) address flow from HRA to IMS | | | | | | | | | | |
| Search | Search-10 | Search recent history | | | | | | | | | | |
| Search | Search 11 | Search on different first name & lastname gives lastname | | | | | | | | | | |
| | | | | | | | | | | | | |
| HRA | HRA-1 | Edit patient profile on search page | | | | | | | | | | |
| | HRA-2 | Verify all HRA data flowed to IMS | | | | | | | | | | |

Summary | Bugs | IMS-Search | Interaction | Quick Actions | HRA | Personal Health History ...

## Assignment No.-4

**AIM:-** Create Defect Report for Any application or web application

**Theory:**

Sample Bug Report

The Sample, Bug/Defect Report given below will give you an exact idea of how to report a Bug in the BugTracking Tool?

Here is an Example scenario that caused a Bug:

Let's assume that in your application under test you want to create a new user with user information, for that you need to login into the application and navigate to the USERS menu -> New User, then enter all the details in the 'User form' like, First Name, Last Name, Age, Address, Phone etc.

Once you enter all this information, you need to click on the 'SAVE' button in order to save the user. Now you can see a successful message saying, "New User has been created successfully".

But when you entered into your application by logging in and you have navigated to the USERS menu -> Newuser, entered all the required information to create the new user and clicked on SAVE button.

BANG! The application crashed and you got one error page on the screen. (Capture this error message windowand save it as a Microsoft paint file)

Now, this is a Bug scenario and you would like to report this as a BUG in your Bug-Tracking Tool.How Will You Report This Bug Effectively?

Sample Bug Report

Here is a sample Bug Report for the above-mentioned example:

(Note that some 'Bug Report' fields might differ depending on your bug tracking system)

SAMPLE BUG REPORT

**Bug Name:** Application crashes upon clicking the SAVE button while creating a new user. **Bug ID:** (It will be automatically created by the BUG Tracking tool once you save this bug). **Area Path:** USERS menu -> New Users

**Build Number:** Version Number 5.0.1 **Severity:** HIGH (High/Medium/Low) or 1 **Priority:** HIGH (High/Medium/Low) or 1**Assigned to:** Developer-X

**Reported By:** Your Name**Reported On:** Date **Reason:** Defect

**Status:** New/Open/Active (Depends on the Tool you are using)

**Environment:** Windows 2003/SQL Server 2005

**Description:** Application crashes upon clicking the SAVE button while creating a new the user, hence unableto create a new user in the application.

Steps to Reproduce:

**1)** Login into the Application.

**2)** Navigate to the Users Menu -> New User

**3)** Filled out all the user information fields.

**4)** Clicked on the 'Save' button.

**5)** Seen an error page "ORA1090 Exception: Insert values Error…"

**6)** See the attached logs for more information (Attach more logs related to the bug..IF any)

**7)** Also see the attached screenshot of the error page.

**Expected Result:** On clicking the SAVE button, you should be prompted to a successful message "New Userhas been created successfully".

(Attach 'application crash' screenshot. IF any)

**Save** the Defect/Bug in the BUG TRACKING TOOL. You will get a Bug ID that you can use for further bugreference.

Default 'New Bug' mail will go to the respective developer and the default module owner (Team leader ormanager) for further action.

How To Write A Good Bug Report? Tips And Tricks

Why a good Bug Report?

If your Bug report is effective, then its chances of getting fixed are higher. So fixing a bug depends upon howeffectively you report it. Reporting a bug is nothing but a skill and in this tutorial we will explain how to achieve this skill.

"The point of writing a problem report (bug report) is to get bugs fixed" – By Cem Kaner. If a tester is notreporting a bug correctly, then the programmer will most likely reject this bug stating it as irreproducible.

This can hurt the tester's morals and sometimes the ego too. (I suggest not to keep any type of ego. ego's like "Ihave reported the bug correctly", "I can reproduce it", "Why has he/she rejected the bug?", "It's not my fault" etc.,).

Qualities of a Good Software Bug Report

Anyone can write a Bug report. But not everyone can write an effective Bug report. You should be able to distinguish between an average bug report and a good bug report.

**How to distinguish between a good and bad Bug Report?** It's very simple, apply the followingcharacteristics and techniques to report a bug.

Characteristics and Techniques

**#1) Having a clearly specified Bug Number:** Always assign a unique number to each bug report. This, in turn, will help you identify the bug record. If you are using any automated bug-reporting tool then this unique number will be generated automatically each time you report a bug.

Note the number and a brief description of each bug that you reported.

**#2) Reproducible:** If your bug is not reproducible, then it will never get fixed.

You should clearly mention the steps to reproduce the bug. Do not assume or skip any reproducing steps. The bug which is described Step by step is easy to reproduce and fix.

**#3) Be Specific:** Do not write an essay about the problem.

Be Specific and to the point. Try to summarize the problem in minimum words yet in an effective way. Do not combine multiple problems even if they seem to be similar. Write different reports for each problem.

Effective Bug Reporting

Bug reporting is an important aspect of Software Testing. Effective Bug reports communicate well with thedevelopment team to avoid confusion or miscommunication.

Good Bug report should be **clear and concise** without any missing key points. Any lack of clarity leads to misunderstanding and slows down the development process as well. Defect writing and reporting is one of themost important but neglected areas in the testing life cycle.

Good writing is very important for filing a bug. The most important point that a tester should keep in mind

is **not to use a commanding tone** in the report. This breaks morale and creates an unhealthy work relationship.Use a suggestive tone.

**Don't assume** that the developer has made a mistake and hence you can use harsh words. Before reporting, it isequally important to check if the same bug has been reported or not.

**A duplicate bug** is a burden in the testing cycle. Check out the whole list of known bugs. At times, the developers may be aware of the issue and ignore it for future releases. Tools like Bugzilla, which automaticallysearches for duplicate bugs, can also be used. However, it is best to manually search for any duplicate bug.

The import information that a bug report must communicate is **"How?" and "Where?"** The report shouldclearly answer exactly how the test was performed and where the defect occurred. The reader should easily reproduce the bug and find out where the bug is.

Keep in mind that the **objective of writing a Bug report** is to enable the developer to visualize the problem. He/She should clearly understand the defect from the Bug report. Remember to provide all the relevant information that the developer is seeking.

Also, bear in mind that a bug report would be preserved for future use and should be well written with the required information. **Use meaningful sentences and simple words** to describe your bugs. Don't use confusingstatements that waste the time of the reviewer.

Report each bug as a separate issue. In case of multiple issues in a single Bug report, you can't close it unless allthe issues are resolved.

Hence, it is best to **split the issues into separate bugs**. This ensures that each bug can be handled separately. A well-written bug report helps a developer to reproduce the bug at their terminal. This will help them diagnose the issue as well.

How To Report A Bug?

**Use the following simple Bug report template:**

This is a simple Bug report format. It may vary depending upon the Bug report tool that you are using. If youare writing a bug report manually then some fields need to be mentioned specifically like the Bug number – which should be assigned manually.

**Reporter:** Your name and email address.

**Product:** In which product you found this bug.**Version:** The product version, if any.

**Component:** These are the major sub-modules of the product.

**Platform:** Mention the hardware platform where you found this bug. The various platforms like 'PC', 'MAC','HP', 'Sun' etc.

**Operating system:** Mention all the operating systems where you found the bug. Operating systems like Windows, Linux, Unix, SunOS, and Mac OS. Also, mention the different OS versions like Windows NT, Windows 2000, Windows XP etc, if applicable.

**Priority:** When should a bug be fixed? Priority is generally set from P1 to P5. P1 as "fix the bug with thehighest priority" and P5 as " Fix when time permits".

**Severity:** This describes the impact of the bug.

Types of Severity:

- **Blocker:** No further testing work can be done.
- **Critical:** Application crash, Loss of data.
- **Major:** Major loss of function.
- **Minor:** Minor loss of function.
- **Trivial:** Some UI enhancements.
- **Enhancement:** Request for a new feature or some enhancement in the existing one.

**Status:** When you are logging the bug into any bug tracking system then by default the bug status will be'New'.

Later on, the bug goes through various stages like Fixed, Verified, Reopen, Won't Fix, etc.

**Assign To:** If you know which developer is responsible for that particular module in which the bug occurred,then you can specify the email address of that developer. Else keep it blank as this will assign the bug to the module owner, if not the Manager will assign the bug to the developer. Possibly add the manager's email address to the CC list.

**URL:** The page URL on which the bug occurred.

**Summary:** A brief summary of the bug, mostly within 60 words or below. Make sure your summary isreflecting on what the problem is and where it is.

**Description:** A detailed description of the bug.

Use the following fields for the description field:

- **Reproduce steps:** Clearly, mention the steps to reproduce the bug.
- **Expected result:** How the application should behave on the above-mentioned steps.
- **Actual result:** What is the actual result of running the above steps i.e. the bug behavior.

These are the important steps in the bug report. You can also add "Report Type" as one more field which willdescribe the bug type.

Report Types include:

1) Coding error
2) Design error
3) New Suggestion
4) Documentation issue
5) Hardware problem

Important Features in Your Bug Report

**Given below are the important features in the Bug report:**

*#1) Bug Number/id*

A Bug number or an identification number (like swb001) makes bug reporting and the process of referring to bugs much easier. The developer can easily check if a particular bug has been fixed or not. It makes the whole testing and retesting process smoother and easier.

*#2) Bug Title*

Bug titles are read more often than any other part of the bug report. This should explain all about what comes in the bug. The Bug title should be suggestive enough that the reader can understand it. A clear bug title makes it easy to understand and the reader can know if the bug has been reported earlier or has been fixed.

*#3) Priority*

Based on the severity of the bug, a priority can be set for it. A bug can be a Blocker, Critical, Major, Minor, Trivial, or a suggestion. Bug priorities can be given from P1 to P5 so that the important ones are viewed first.

*#4) Platform/Environment*

OS and browser configuration is necessary for a clear bug report. It is the best way to communicate how the bugcan be reproduced.

Without the exact platform or environment, the application may behave differently and the bug at the tester's end may not replicate on the developer's end. So it is best to clearly mention the environment in which the bugwas detected.

*#5) Description*

Bug description helps the developer to understand the bug. It describes the problem encountered. A poor description will create confusion and waste the time of the developers as well as testers.

It is necessary to communicate clearly about the effect of the description. It's always helpful to use complete sentences. It is a good practice to describe each problem separately instead of crumbling them altogether. Don't use terms like "I think" or "I believe".

*#6) Steps to Reproduce*

A good Bug report should clearly mention the steps to reproduce. These steps should include actions that maycause the bug. Don't make generic statements. Be specific on the steps to follow.

A good example of a well-written procedure is given belowSteps:

- Select product Abc01.
- Click on Add to cart.
- Click Remove to remove the product from the cart.

*#7) Expected and Actual Result*

A Bug description is incomplete without the Expected and Actual results. It is necessary to outline what the outcome of the test is and what the user should expect. The reader should know what the correct outcome of thetest is. Clearly, mention what happened during the test and what the outcome was.

*#8) Screenshot*

A picture is worth a thousand words. Take a Screenshot of the instance of failure with proper captioning to highlight the defect. Highlight unexpected error messages with light red color. This draws attention to the required area.

Some Bonus Tips To Write A Good Bug Report

Given below are some additional tips on how to write a good Bug report:

#1) Report the problem immediately

If you find any bugs while testing, then you do not need to wait to write a detailed bug report later. Instead, write a bug report immediately. This will ensure a good and reproducible Bug report. If you decide to write the Bug report later on then there is a higher chance to miss the important steps in your report.

#2) Reproduce the bug three times before writing a Bug report

Your bug should be reproducible. Make sure that your steps are robust enough to reproduce the bug without any ambiguity. If your bug is not reproducible every time, then you can still file a bug mentioning the periodic nature of the bug.

#3) Test the same bug occurrence on other similar modules

Sometimes the developer uses the same code for different similar modules. So there is a higher chance for the bug in one module to occur in other similar modules as well. You can even try to find the more severe version of the bug you found.

#4) Write a good bug summary

Bug summary will help the developers to quickly analyze the bug's nature. A poor quality report will unnecessarily increase development and testing time. Communicate well with your bug report summary. Keep in mind that the bug summary can be used as a reference to search for the bug in the bug inventory.

#5) Read the Bug report before hitting the Submit button

Read all the sentences, wordings and steps that are used in the bug report. See if any sentence is creating ambiguity that can lead to misinterpretation. Misleading words or sentences should be avoided in order to have a clear bug report.

#6) Do not use abusive language.

It's nice that you did a good work and found a bug but do not use this credit for criticizing the developer or to attack any individual.

Sample Bug Reports For Web And Product Applications

**Bug report sample 1: Web Project bug report**

**Summary:** In CTR (Click through ratio) 'Total' row calculation is wrong

**Product:** Example product

**Version:** 1.0

Platform: **PC**

**URL:** (Provide url of page where bug occurs)

**OS/Version:** Windows 2000

**Component:** Publisher stats

**Assigned To:** developer@example.com

**Reported By:** tester@example.com **CC:** manager@example.com

Bug Description:

**Reproduce steps:**

1) Go to page: (Provide URL of page where bug occurs)
2) Click on 'Publisher stats' link to view publisher's revenue detail stats date wise.
3) On page (Provide URL of page where bug occurs) check CTR value in 'Total' row of CTR stats table. Actual result: Calculation of 'Total' row in CTR table is wrong. Also Individual row CTR for each publisher isnot truncated to 2 digits after decimal point. It's showing CTR like 0.042556767

Expected result: Total CTR= (Total clicks/Total searches)*100 [Attach bug screenshot if any]

Please fix the bug.

**********

Sample bug report 2: Application product

Bug report sample Application testing scenario:

Lets assume in your application you want to create a new user with his/her information, for that you need to logon into the application and navigate to USERS menu > New User, then enter all the details in the User form like, First Name, Last Name, Age, Address, Phone etc. Once you enter all these need to click on SAVE button in order to save the user and you can see a success message saying, "New User has been created successfully". Now you entered into your application by logging in and navigate to USERS menu > New user, entered all the information and clicked on SAVE button and now the application crashed and you can see one error page on the

screen, now you would like to report this BUG.

Now here is how we can report bug for above scenario:

**Bug Name:** Application crash on clicking the SAVE button while creating a new user.

**Bug ID:** The BUG Tracking tool will automatically create it once you save this.

**Area Path:** USERS menu > New Users**Build Number:**/Version Number 5.0.1 **Severity:** HIGH (High/Medium/Low) **Priority:** HIGH (High/Medium/Low) **Assigned to:** Developer-X

**Status:** New/Open/Active – Depends on the Tool you are using

**Environment:** Windows 2003/SQL Server 2005

Description:

Application crash on clicking the SAVE button while creating a new user, hence unable to create a new user inthe application.

Steps To Reproduce:

1) Logon into the application
2) Navigate to the Users Menu > New User
3) Filled all the fields
4) Clicked on 'Save' button
5) Seen an error page "ORA1090 Exception: Insert values Error…"
6) See the attached logs for more information
7) And also see the attached screenshot of the error page.

Expected: On clicking SAVE button should be prompted to a success message "New User has been createdsuccessfully".

Save the defect/bug in the BUG TRACKING TOOL.

**Conclusion:** Successfully studied Defect Report for web application

# Assignment No.-5

**AIM:-** Installation of Selenium grid and selenium Web driver & java eclipse (automation tools).

Theory:

What is Selenium?

**Selenium** is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as **Selenium Testing**.

## Selenium Tool Suite

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid

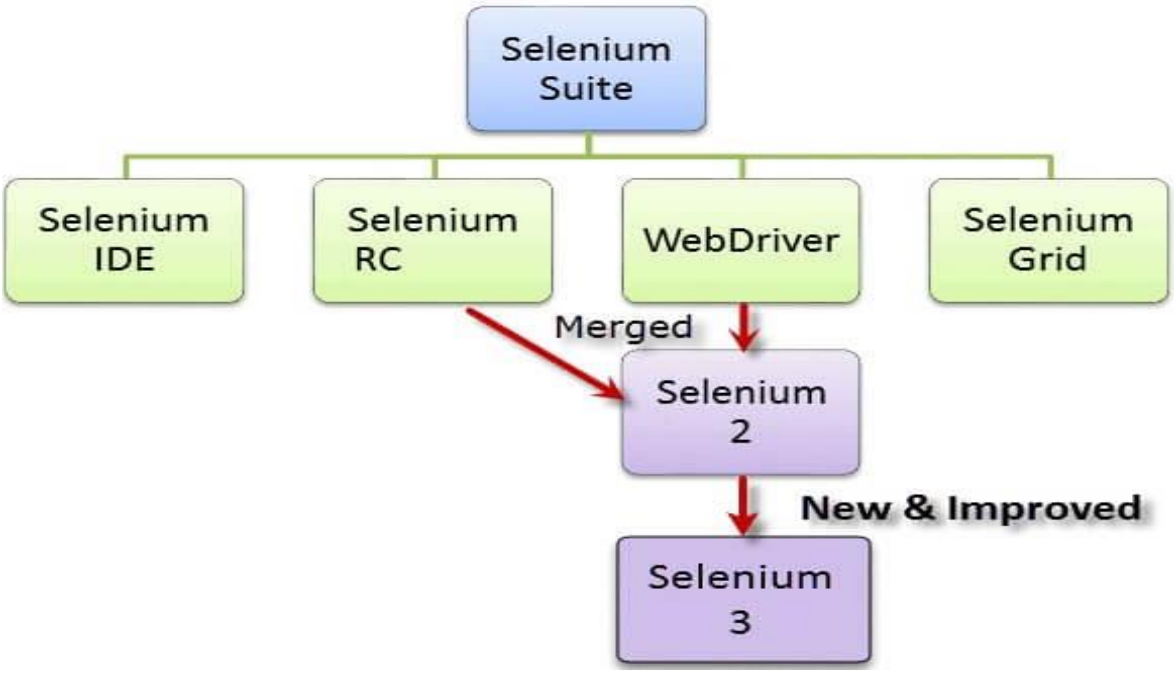
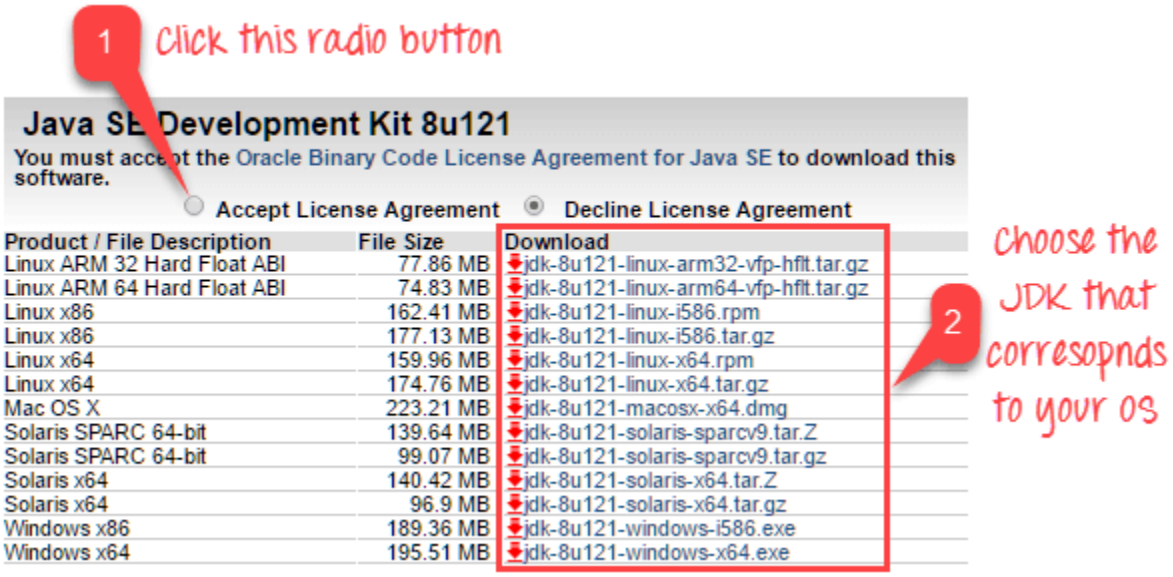



Fig 1. Selenium Suit

# How to Download & Install Selenium WebDriver?

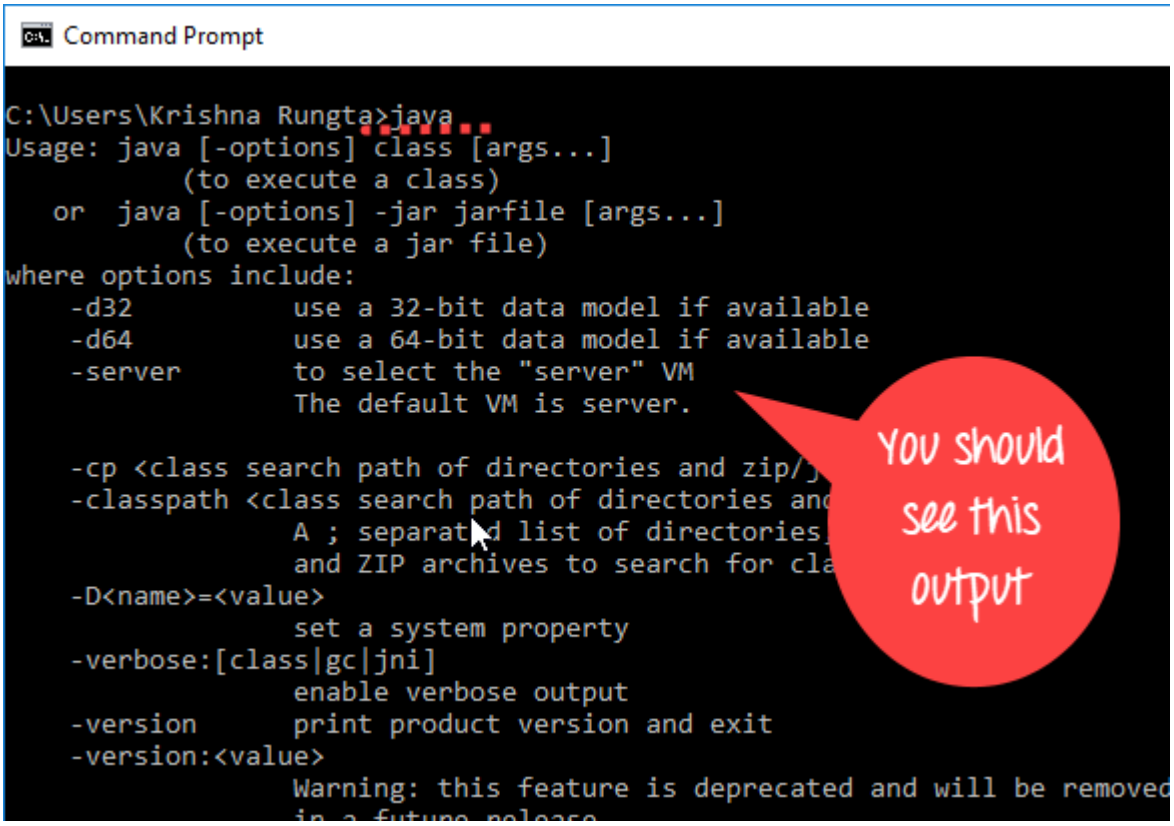
### Step 1 – Install Java on your computer

Download and install the **Java Software Development Kit (JDK)**

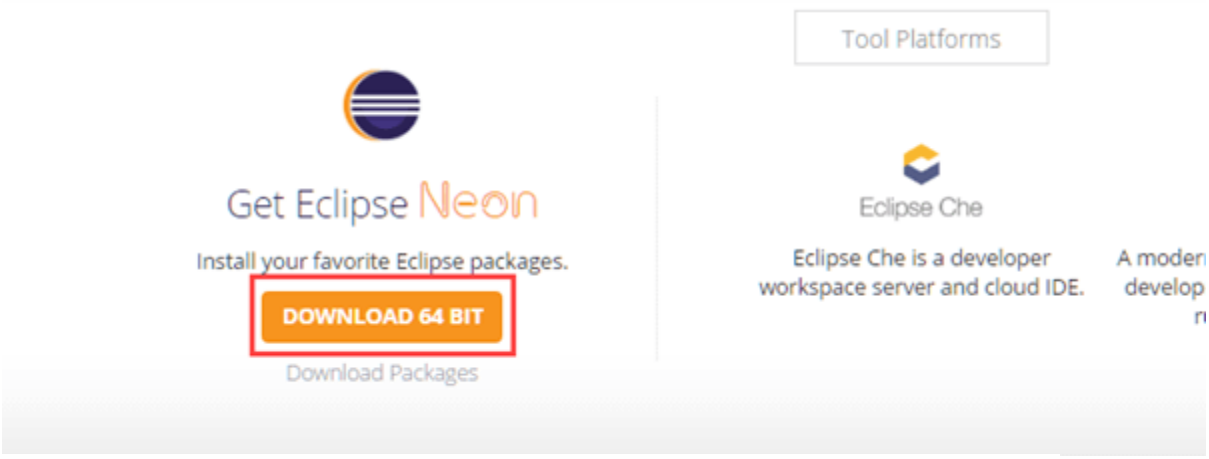
This JDK version comes bundled with Java Runtime Environment (JRE), so you do not need to download and install the JRE separately.

Once installation is complete, open command prompt and type “java”. If you see the following screen you are good to move to the next step

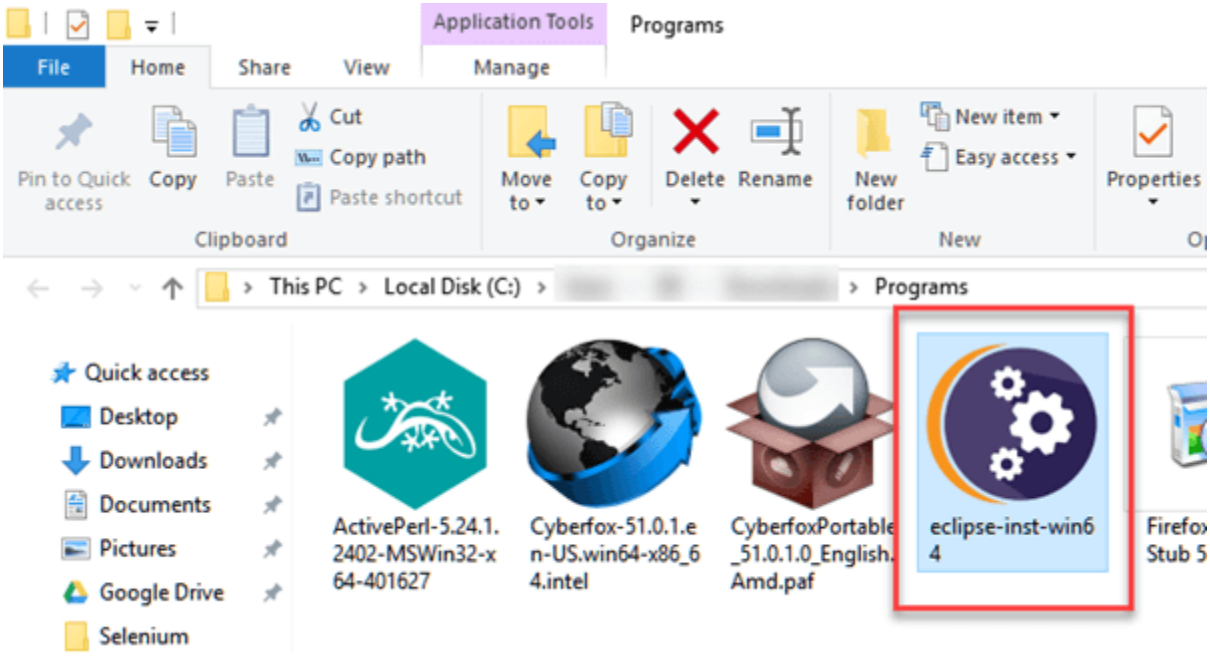



**Step 2 – Install Eclipse IDE**

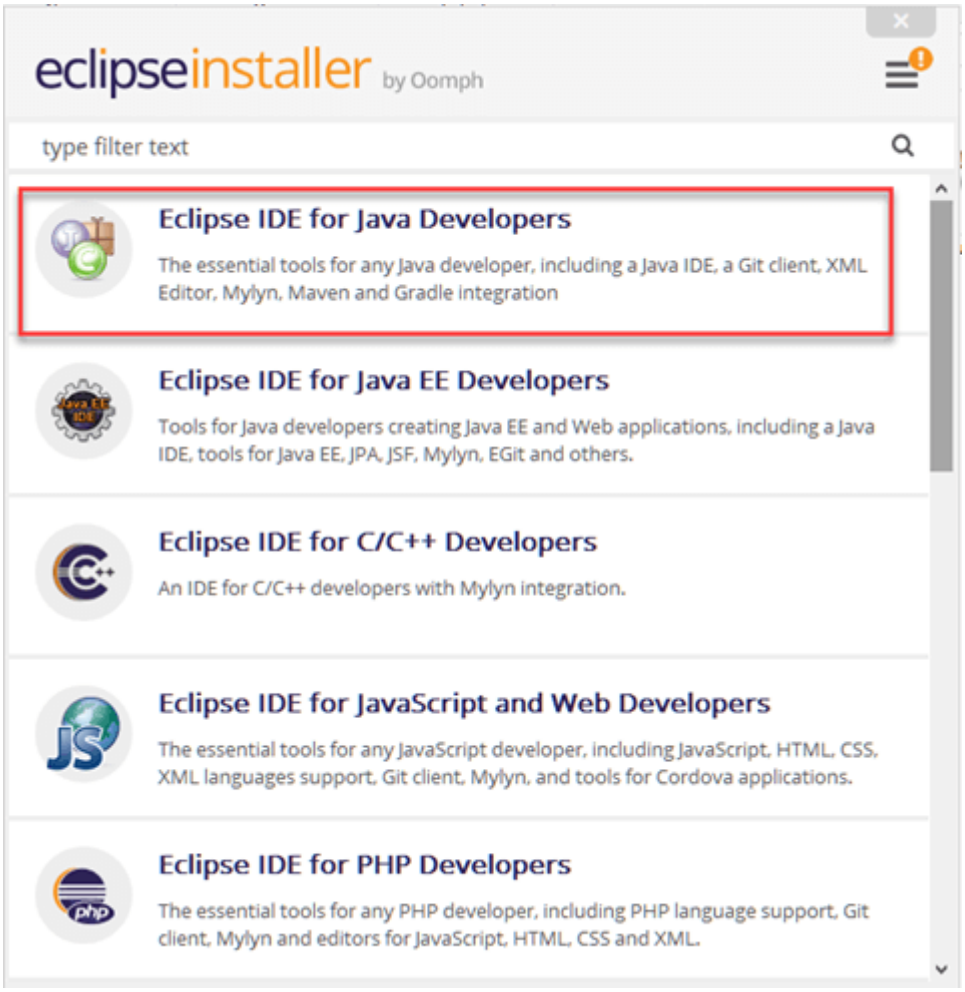
Download latest version of **“Eclipse IDE for Java Developers”** here. Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.



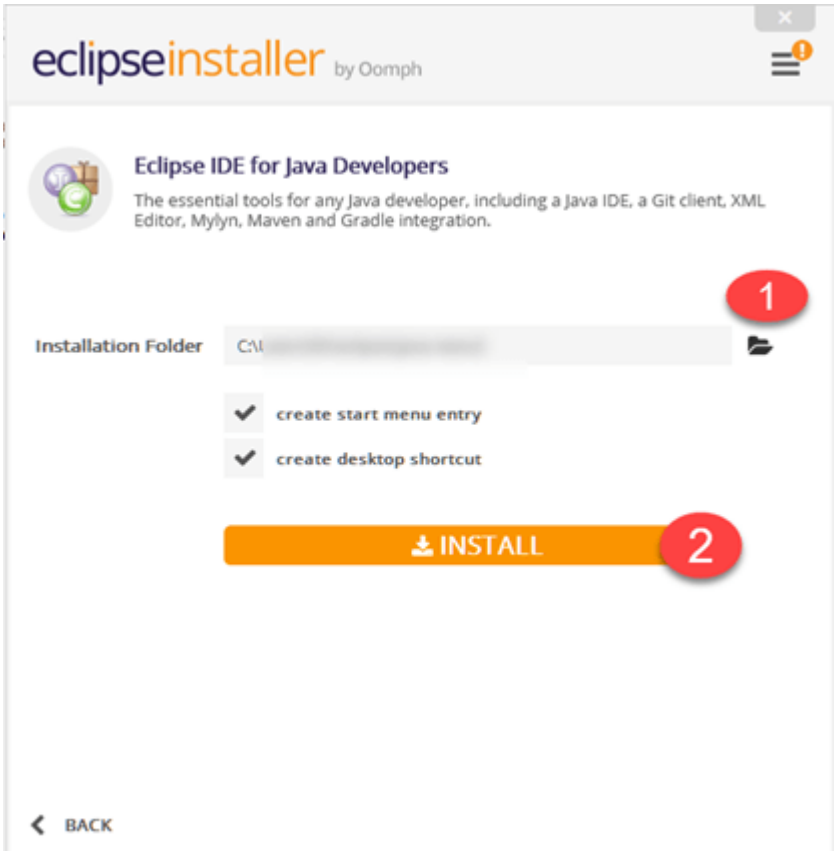

You should be able to download an exe file named “eclipse-inst-win64” for Setup.



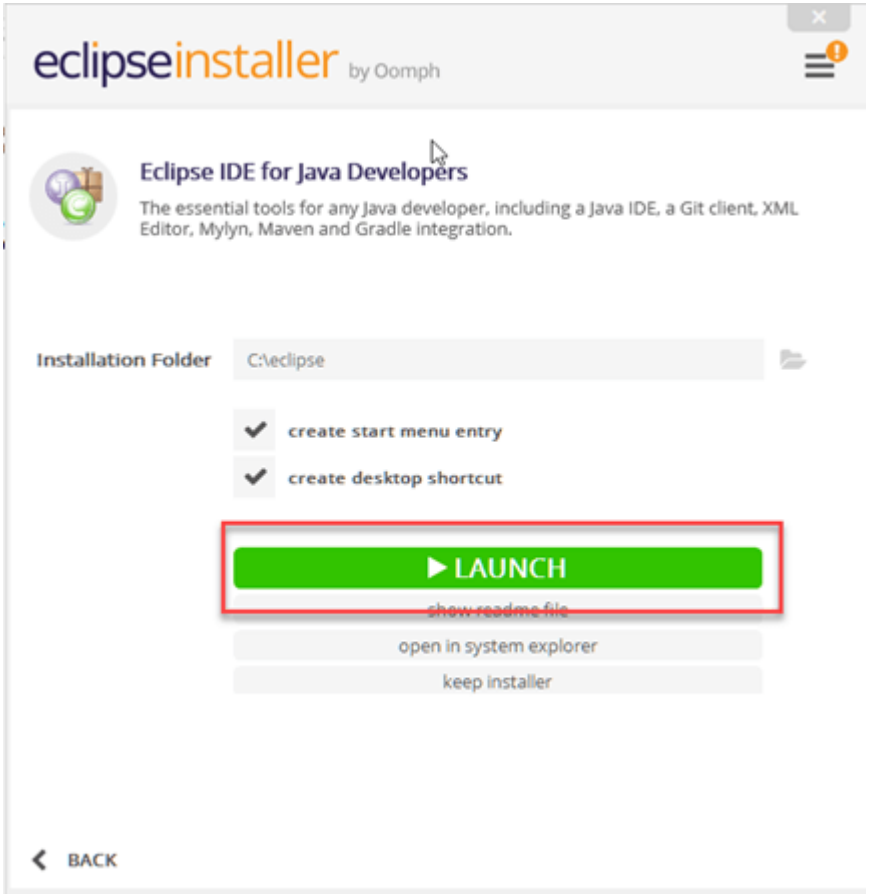

Double-click on file to Install the Eclipse. A new window will open. Click Eclipse IDE for Java Developers.




After that, a new window will open which click button marked 1 and change path to "C:\eclipse". Post that Click on Install button marked 2



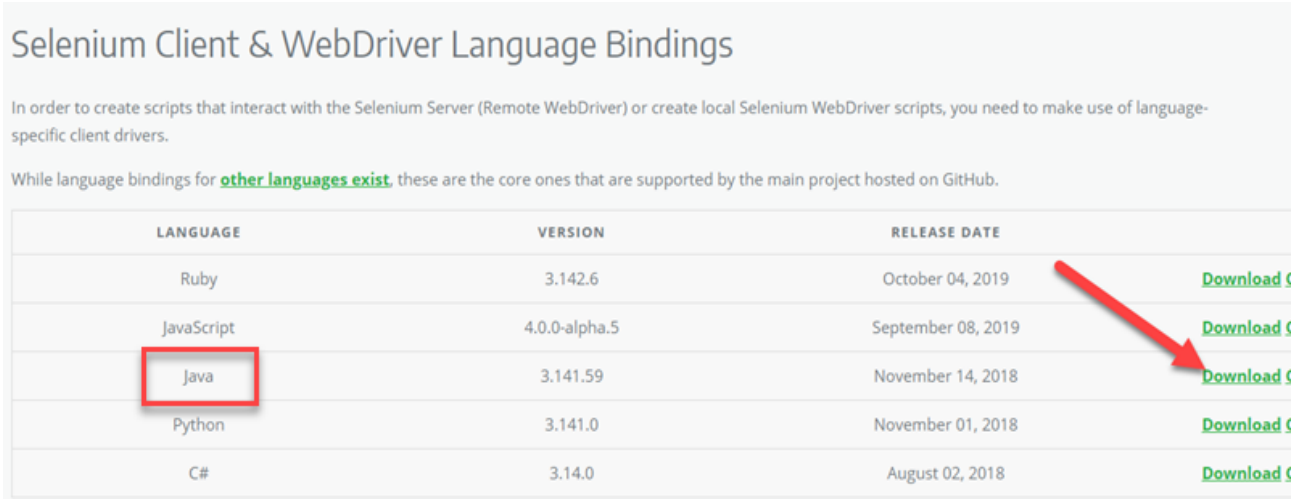

After successful completion of the installation procedure, a window will appear. On that window click on Launch




This will start eclipse neon IDE for you.

**Step 3 – Download the Selenium Java Client Driver**

You can download **Selenium Webdriver for Java Client Driver** here. You will find client drivers for other languages there, but only choose the one for Java.

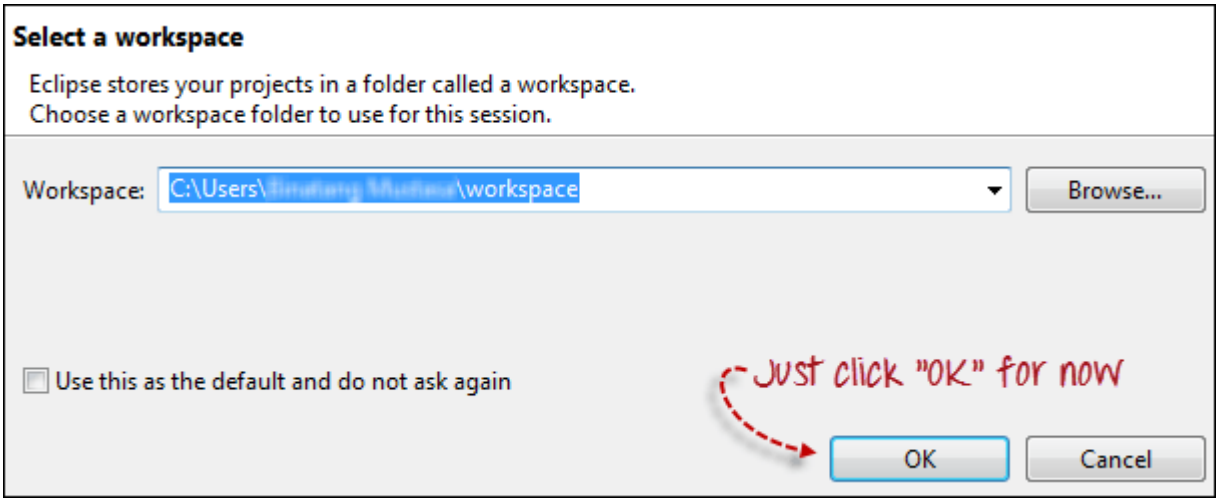


This download comes as a ZIP file named "selenium-3.14.0.zip". For simplicity of Selenium installation on Windows 10, extract the contents of this ZIP file on your C drive so that you would have the directory "C:\selenium-3.14.0\". This directory contains all the JAR files that we would later import on Eclipse for Selenium setup.

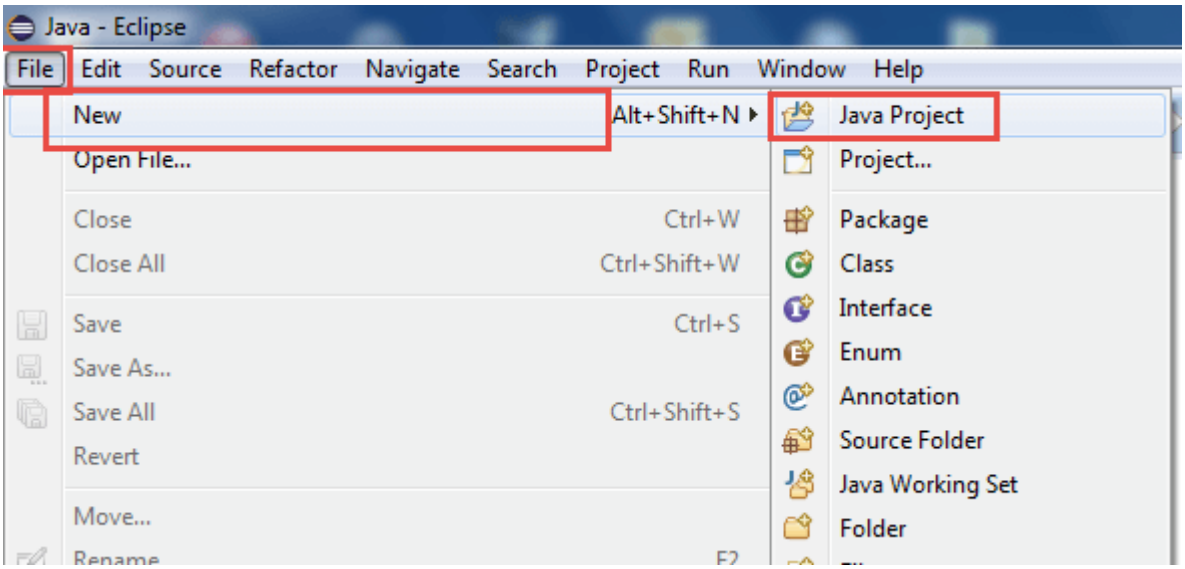
**Step 4 – Configure Eclipse IDE with WebDriver**

1. Launch the "eclipse.exe" file inside the "eclipse" folder that we extracted in step 2. If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.

When asked to select for a workspace, just accept the default location.

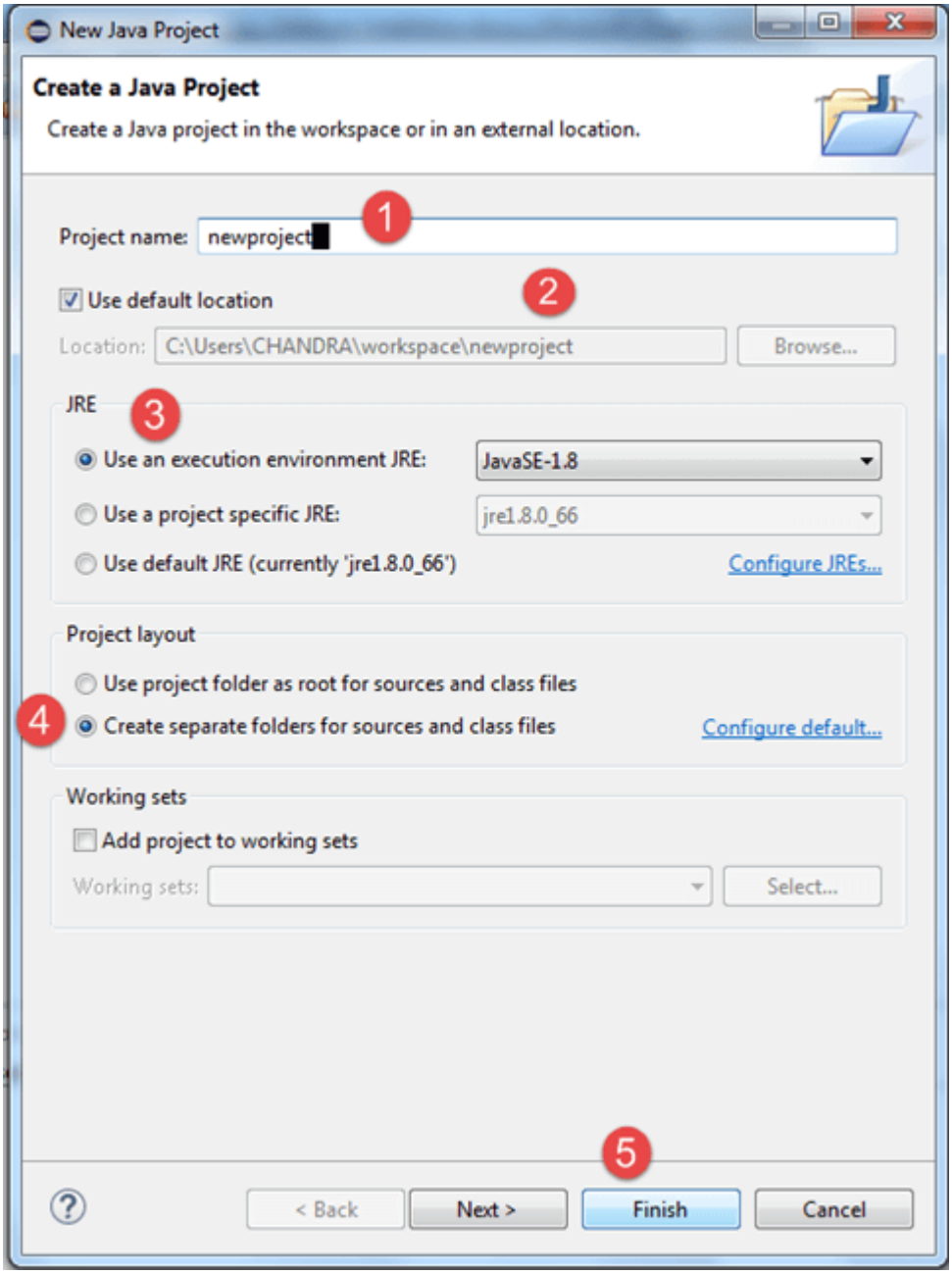



3. Create a new project through File > New > Java Project. Name the project as "newproject".



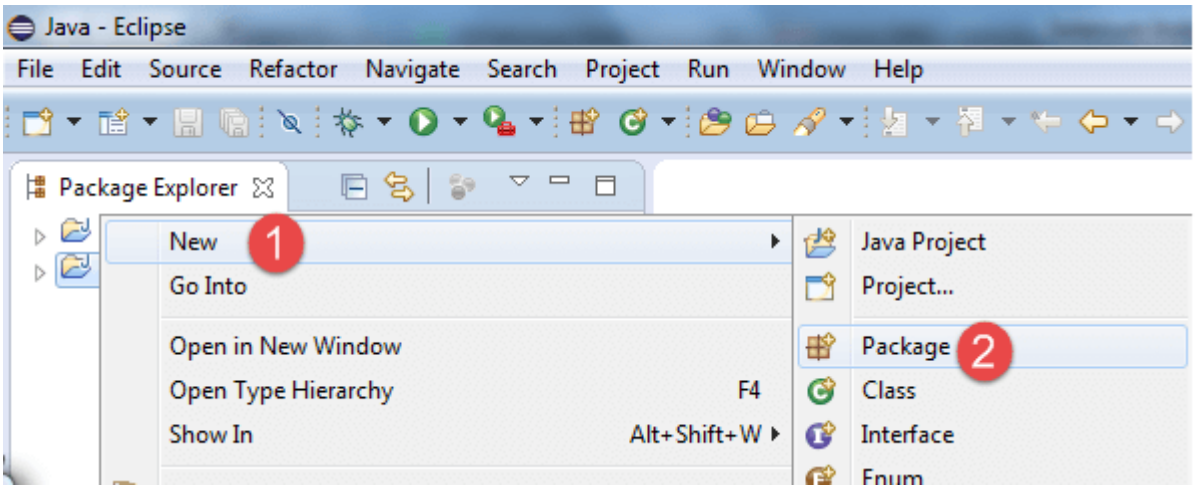

A new pop-up window will open enter details as follow

1. Project Name
2. Location to save project
3. Select an execution JRE
4. Select layout project option
5. Click on Finish button



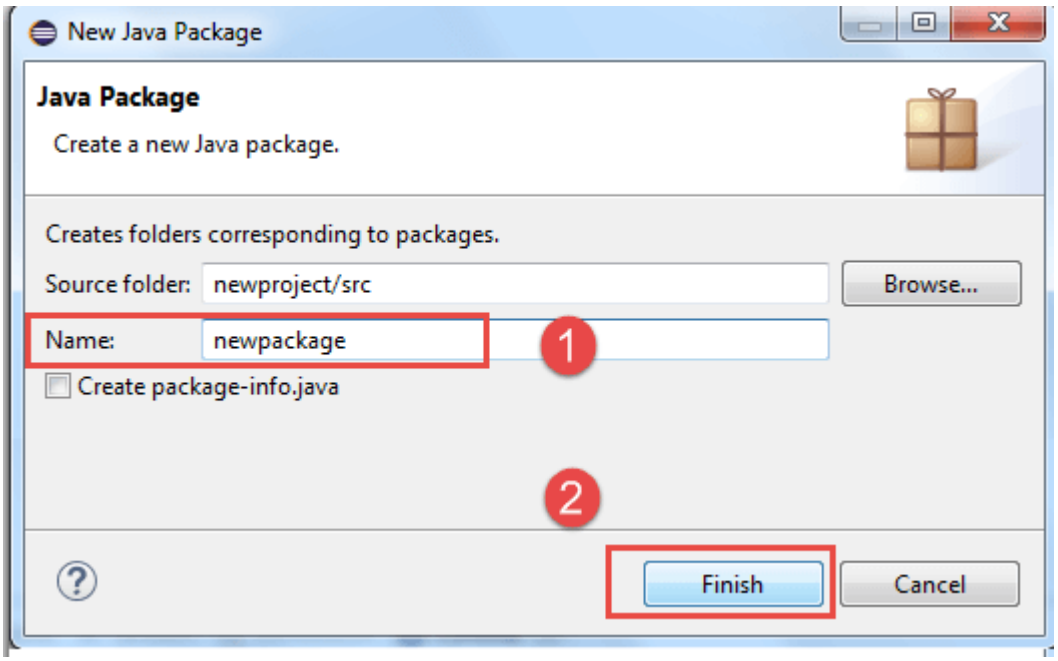

4. In this step,

1. Right-click on the newly created project and
2. Select New > Package, and name that package as “newpackage”.

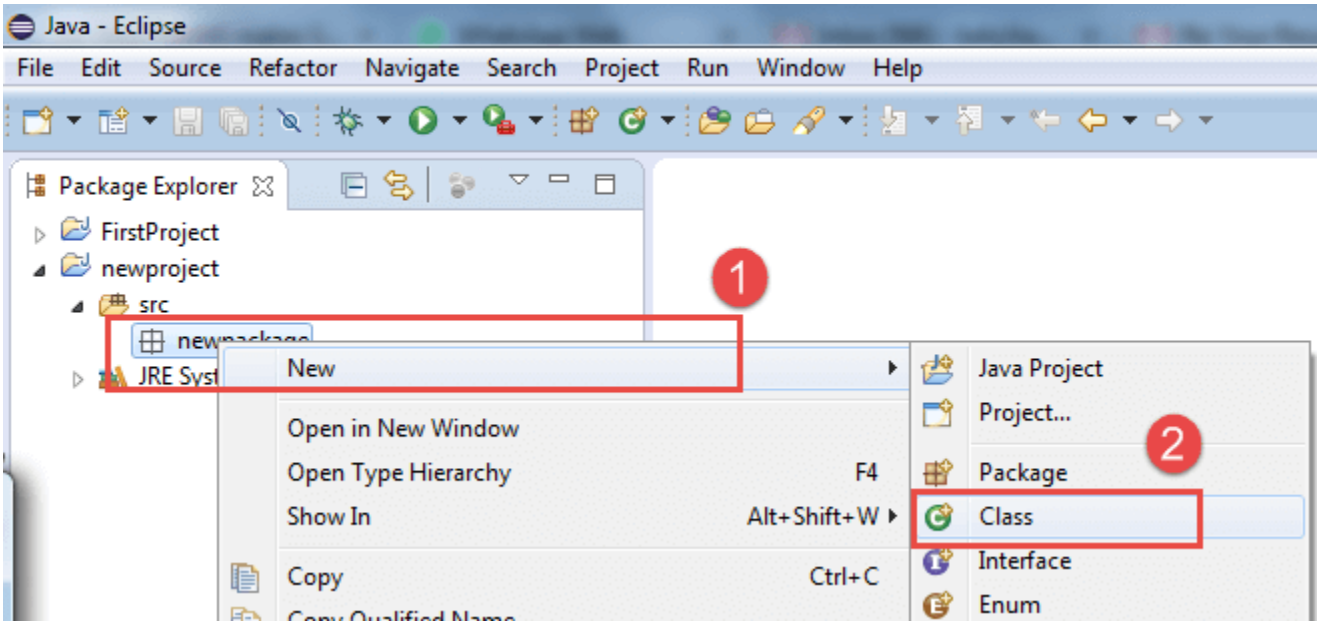



A pop-up window will open to name the package,

1. Enter the name of the package
2. Click on Finish button

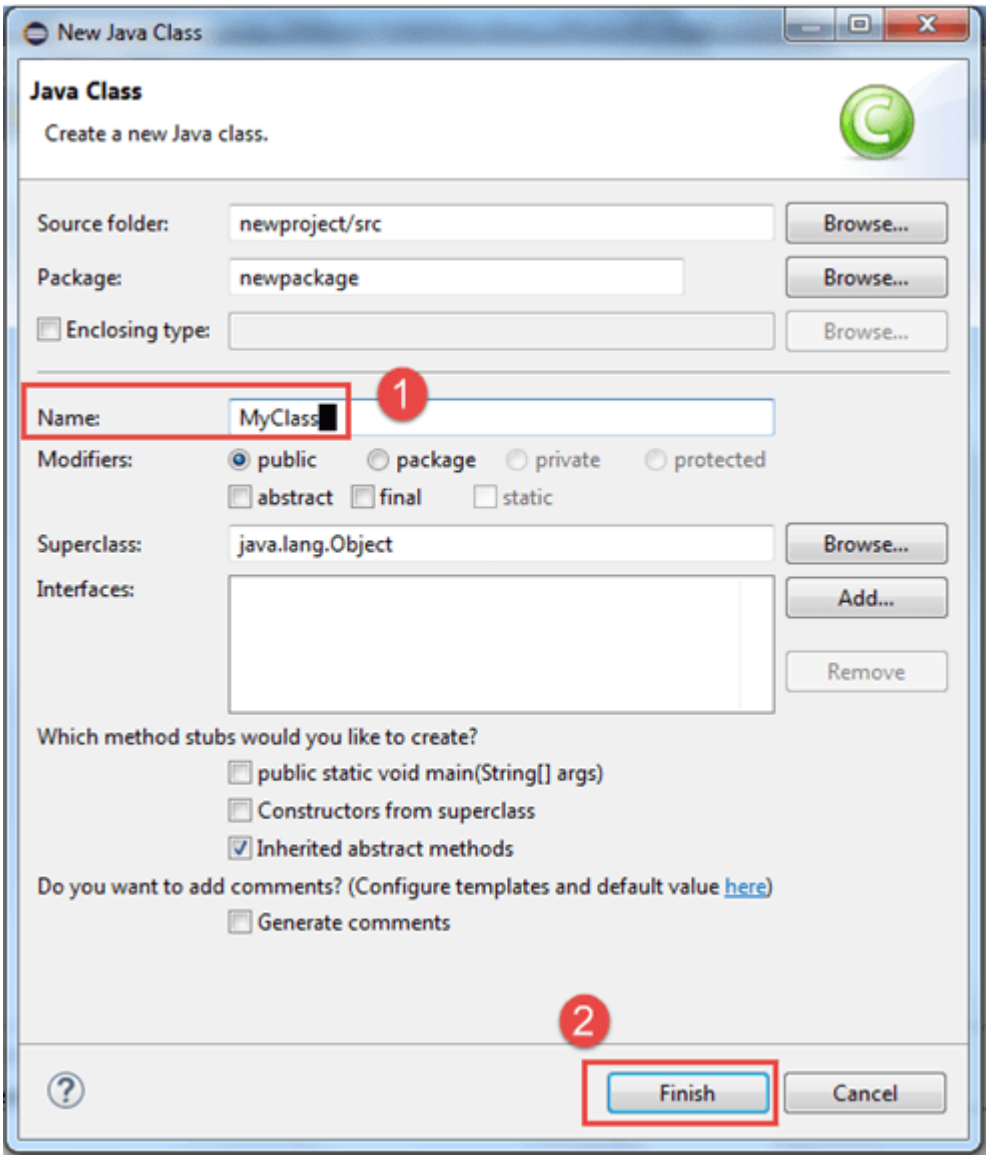



5. Create a new Java class under newpackage by right-clicking on it and then selecting- New > Class, and then name it as “MyClass”. Your Eclipse IDE should look like the image below.

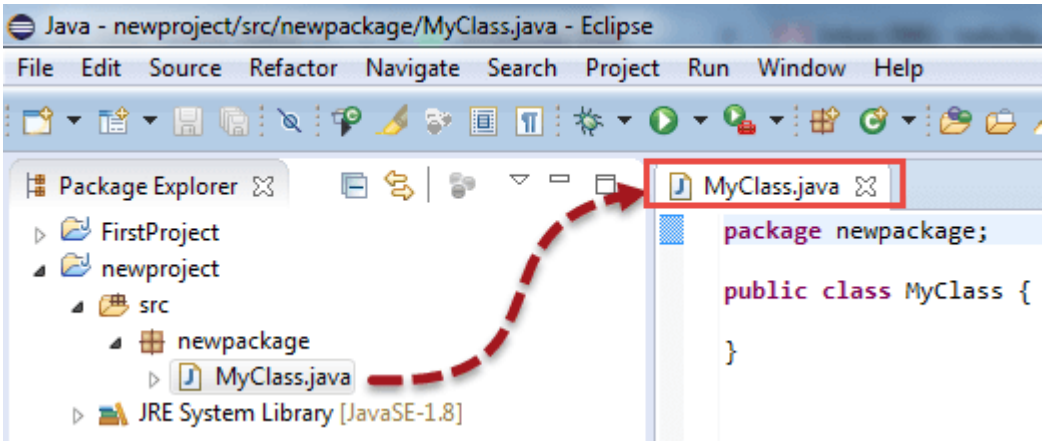



When you click on Class, a pop-up window will open, enter details as

1. Name of the class
2. Click on Finish button



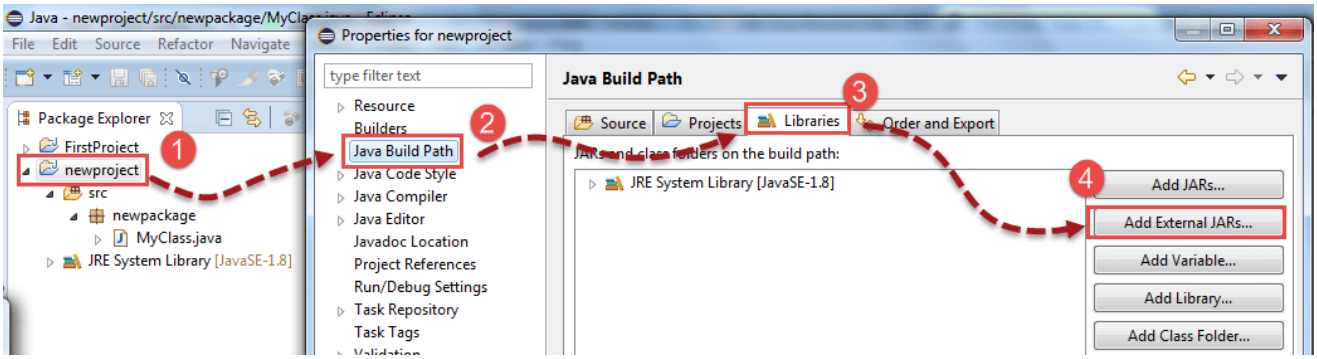

This is how it looks like after creating class.



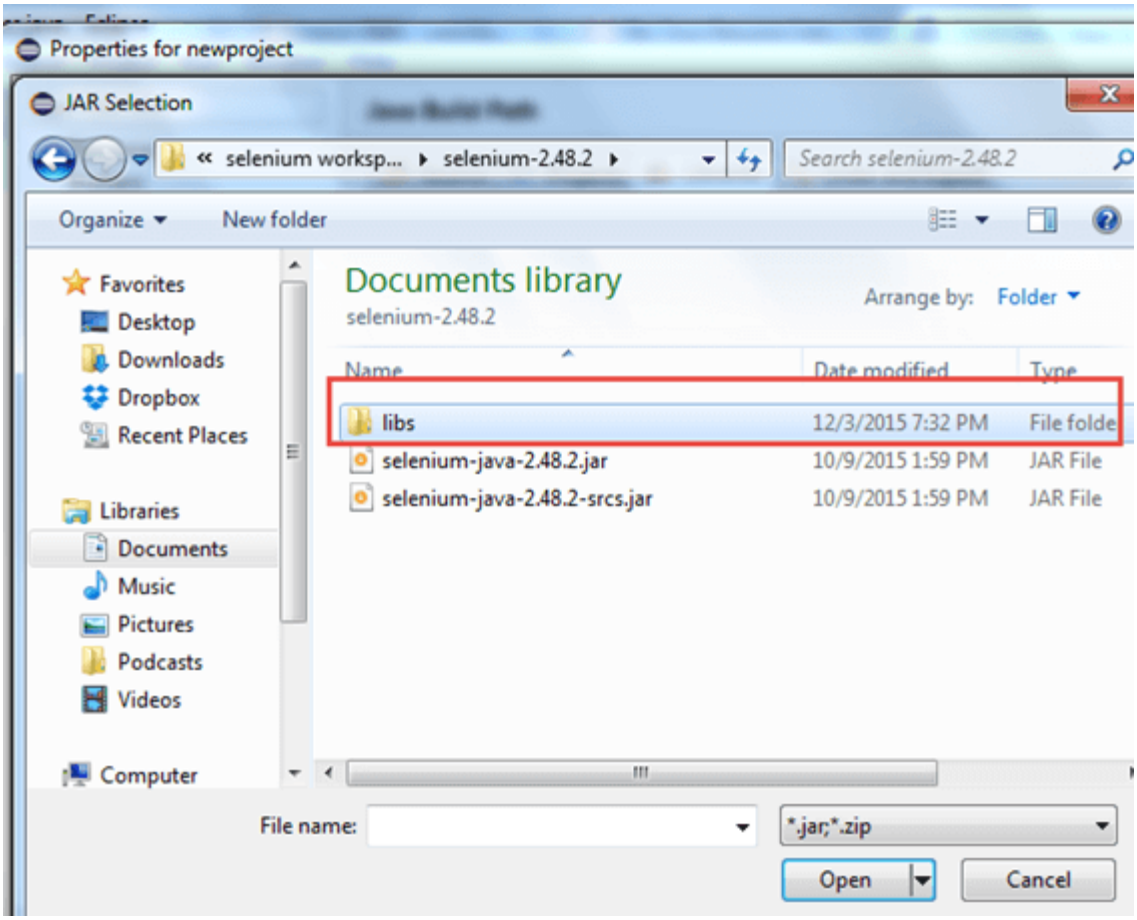

Now selenium WebDriver's into Java Build Path

In this step,

1. Right-click on “newproject” and select **Properties**.
2. On the Properties dialog, click on “Java Build Path”.
3. Click on the **Libraries** tab, and then
4. Click on “Add External JARs..”

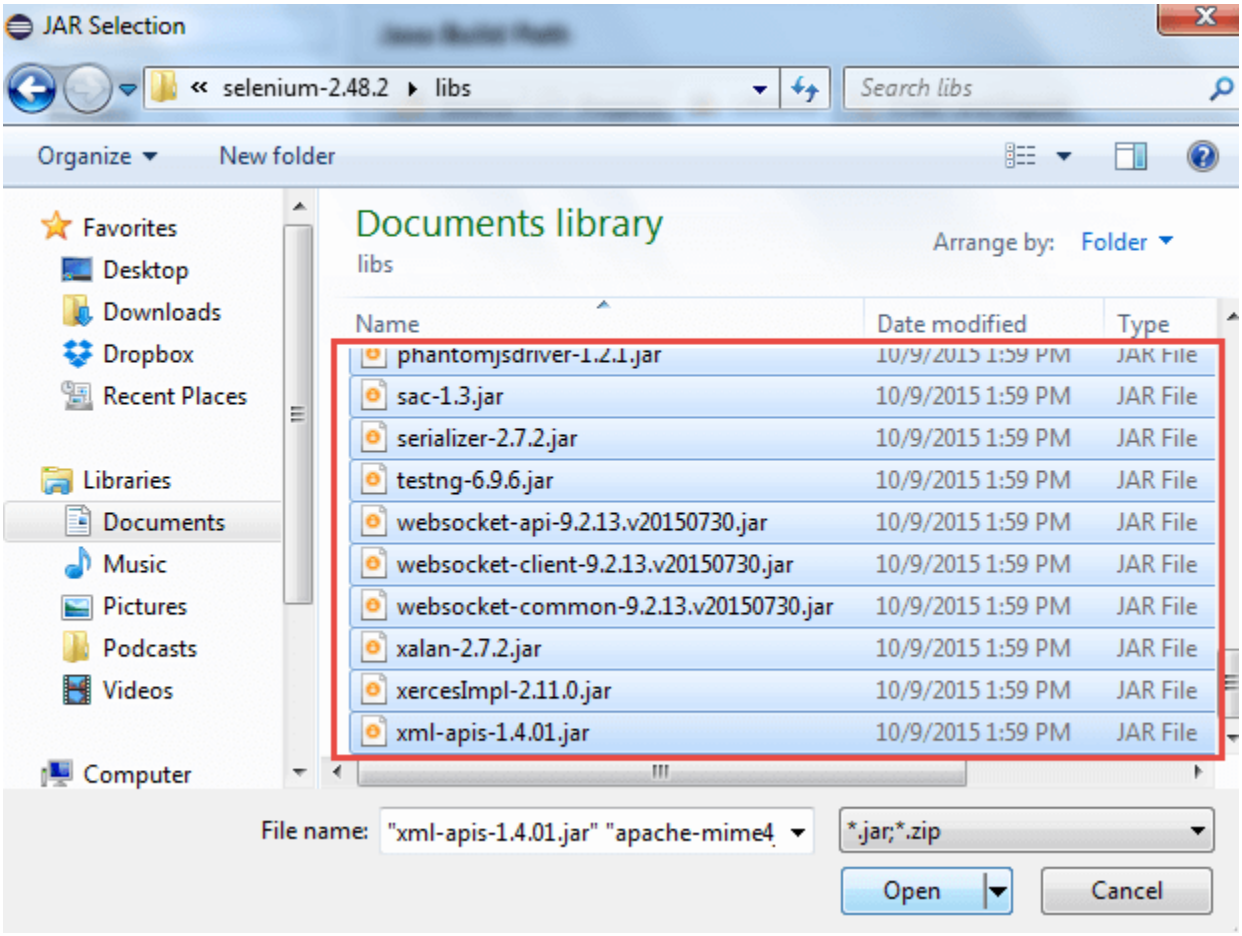



When you click on “Add External JARs..” It will open a pop-up window. Select the JAR files you want to add.

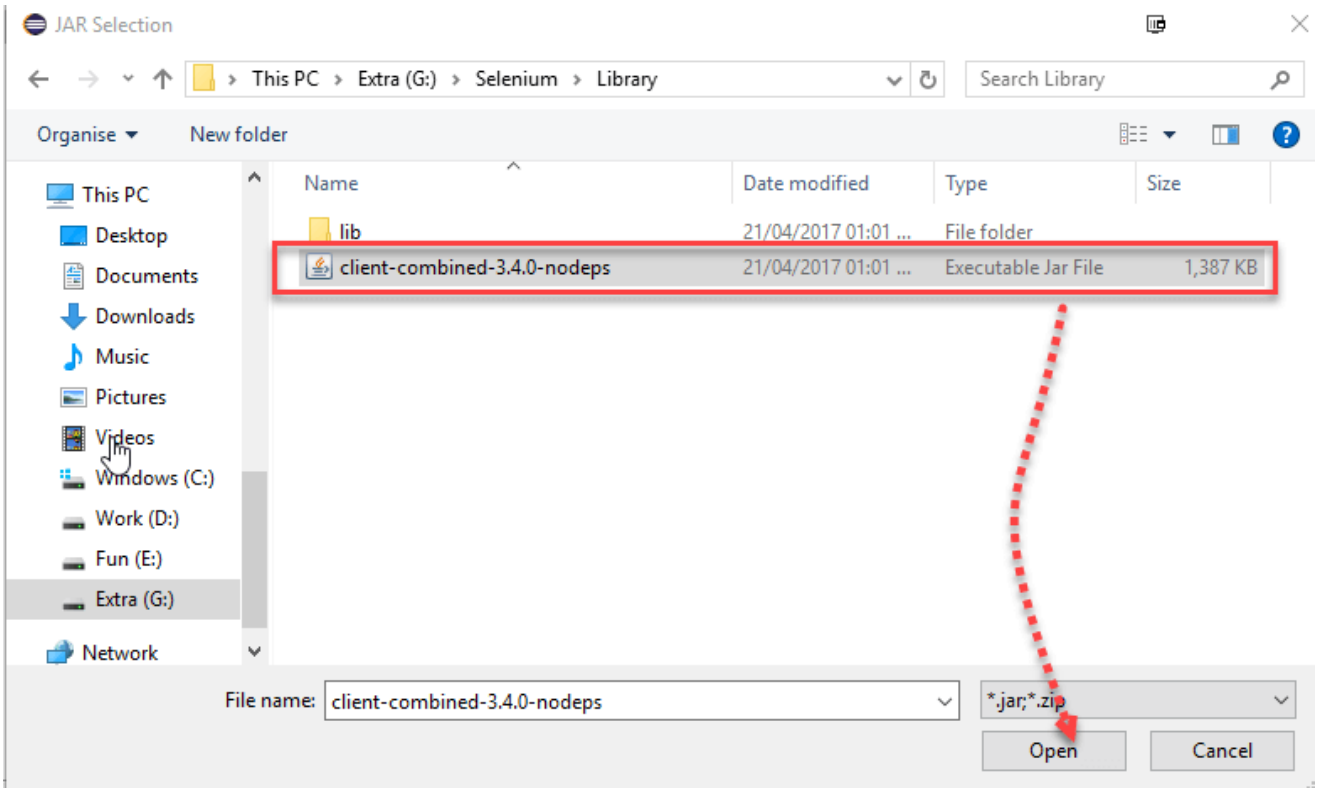



After selecting jar files, click on OK button.

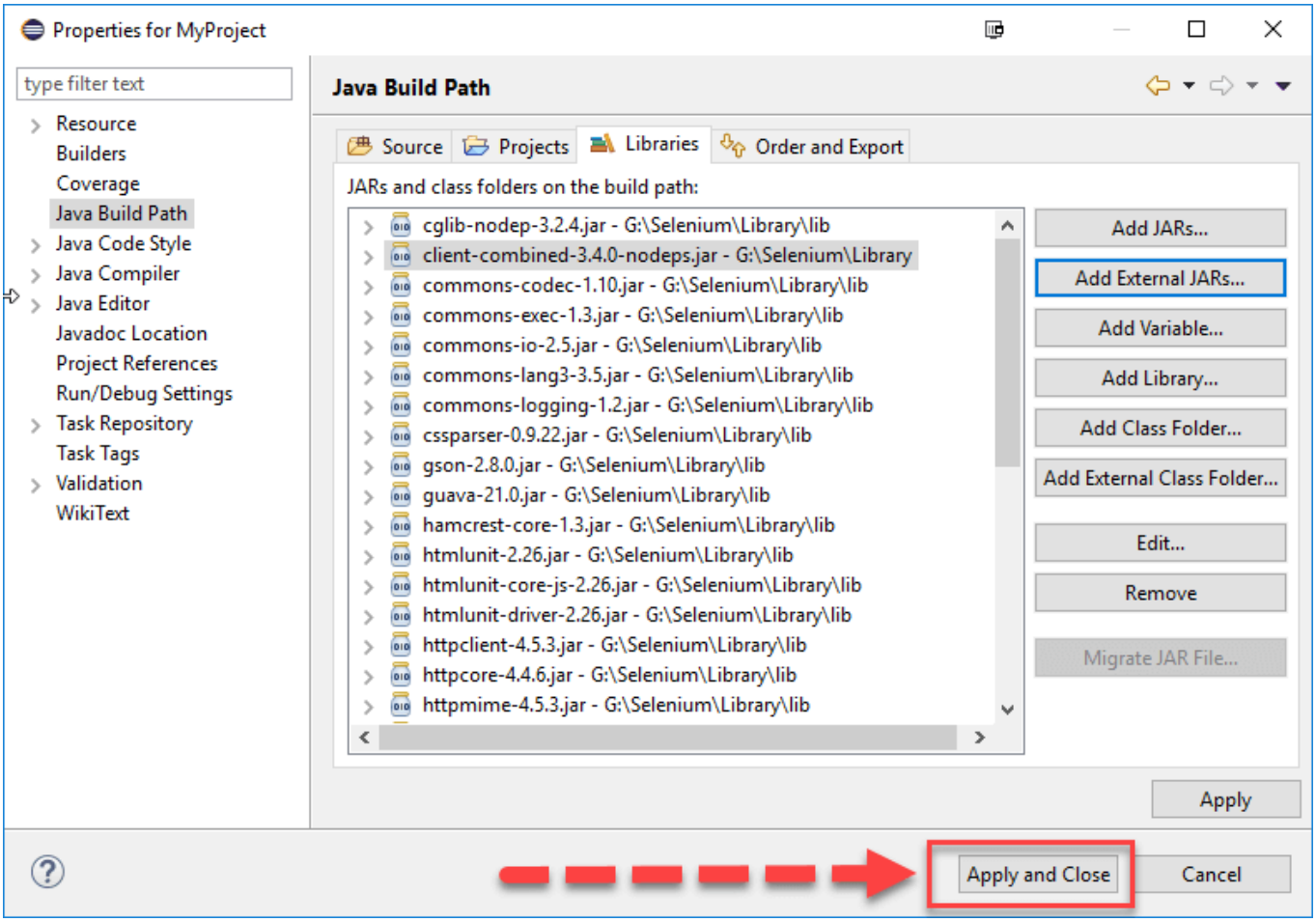
Select all files inside the lib folder.



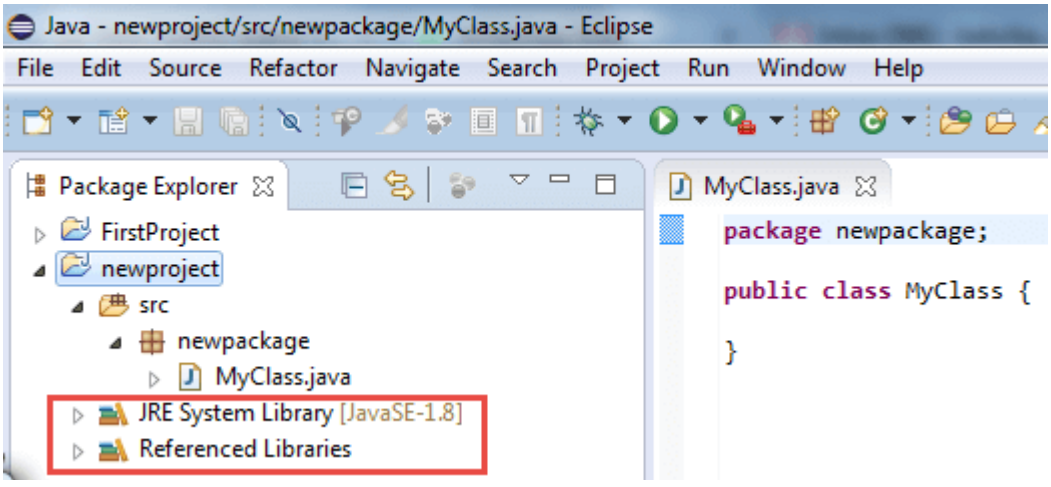

Select files outside lib folder




Once done, click “Apply and Close” button




6. Add all the JAR files inside and outside the "libs" folder. Your Properties dialog should now look similar to the image below.




7. Finally, click OK and we are done importing Selenium libraries into our project.

**Conclusion:** Successfully studied Installation of Selenium grid and selenium Webdriver & java eclipse.