Quiz-1

```
String str = "Scaler Academy"
              0 1 2 3 4

S.O.Pln (str.charAt(4));
```

Ans → e

Quiz 2

```
String str = "Scaler Academy"
              0 1 2 3 4

S.O.P (str.length); // Error
```

length is a function needs ()

Quiz 3.

```
String str = "abcd"

S.O.Pln(str.length()); // 4
```

String		Integer
Haryana	-	10 M
Delhi	-	15 M
Punjab	-	8 M
U.P	-	30 M

	0	1	2	3
States	H	P	D	U.P
Population	10	8	15	30

✓ Ques. find if you have the population of a particular state present or not

✓ Ques. find the population of a state

Delhi — 15 M

population [2]

✓ Ques. update the population of a state

P — 8 → 9

find. index from states [ ]
then update in population [ ]

(key , value)

(State , population) (String , Integer)

↓ ↓

unique

(country , currency)

Keys are unique

Syntax →

HashMap < key data type , value data type > map = new HashMap <>();

map ↓ name

map = { Delhi → 10
Haryana → 9
U.P → 30 }

Basic operation

① find map.containsKey ("Delhi");

↳ true

map.containsKey ("Gujarat");
↳ false

② get the value of a particular key

map.get ("Delhi"); ⟶ 10

//ide map.get ("Gujarat");

map = { Delhi → 6
Haryana → 9
U.P → 30
Gujarat → 7 }

③ Add / update

map.put ("Gujarat", 7); // add when key is not present

map.put ("Delhi", 6); // update when key is already present

④ Size (number of keys)

```
map.size();   // 4
```

Order in HashMap is random

⑤ Print →

```
S.O.P (map);
```

map =

{ Delhi → 6
U.P → 30
Gujarat → 7 }

⑥ Remove →

```
map.remove("Haryana");
```

For-each Loop

int[] arr = | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
0 1 2 3 4 5 6

```
for(int i=0; i< arr.length; i++)
```

arr[i]

name of variable

```
for( int val : arr ) {
```

data type

val = 1

```
    S.O.P (val);
}
```

Output → 1 2 3 4 5 6 7

```
String [] arr = { "Delhi", "Haryana", "Punjab" };
```
0 1 2

```
for( String ele : arr ) {
```

i = 0

arr[0]

ele = Delhi

ele = Haryana

```
    S.O.P (ele);
```

i++

arr[1]

}

Delhi Haryana Punjab

Print all the keys & value using a for loop.

keyset()

```
for ( String key : map.keySet() ) {
    S.O.P (key);  // print keys of your HashMap
    S.O.P (map.get(key))
}
```

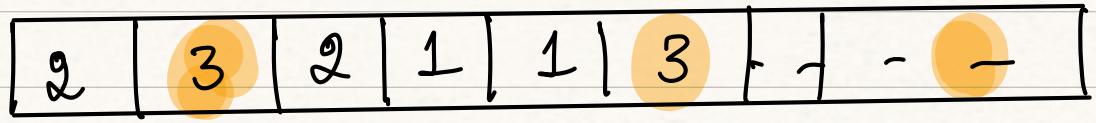
Delhi
U.P.
Gujarat

Delhi - 10 U.P. - 30 Gujarat - 10

Ques. Given an array, calculate the frequency of each element.

arr = { 2, 5, 2, 4, 3, 5, 9, 11, 5, 11 };

element	frequency
2	2
5	3
4	1
3	1
9	1
11	2



↳ Nested for loops
↳ Duplicate entries were there

create a hashmap to store the frequency of all the elements.

```
HashMap<Integer, Integer> freqmap = new HashMap<>();
```

{ }

freqmap.containsKey(key); → true/false

freqmap.get(key); → value of key

freqmap.put(key, value); → add/update

arr = { 2, 5, 2, 4, 3, 5, 9, 11, 5, 11 };

0 1 2 3 4 5 6 7 8 9

freqmap.containsKey(2), → false

↳ freqmap.put(2, 1);

{ 2 = ~~1~~ 2
5 = 1 }

freqmap.containsKey(2); → true

```
int oldfreq = map.get(2); //1
map.put(2, oldfreq+1);
```

unique customer id for a week → use HashSet

HashSet < key >
data type

```
HashSet<String> set = new HashSet<>();
```

arr = { "Ram", "Shyam", "Suman", "Ram", "Sunita", "Shyam" }

H.S = { ~~"Ram"~~, "Shyam", "Suman", "Sunita", "Raman" }

Basic Operations -

can only have unique entries

1. whether Key is present

```
set.contains("Ram");   ⟶ true
set.contains("Pooja"); ⟶ false
```

2. add

```
set.add("Ram");
set.add("Raman");
```

3. size

```
set.size(); // 5
```

4. remove

```
set.remove("Ram");
```

(5) S.O.P (set)

(6) Print using for loop

name of Hashset

```
for(String x : set ){
    S.O.P(x);
}
```

Doubts

Ques 1. Beesly string

p a m

1 p & 1 m 1 a & 1 m

delete 2 p's 2 m's delete 3 a's → 3 m's

~~m m m m m a p a p a p a m m~~

total of 3 p's & 4 a's ⟶ 7 m's

return 1 [count of p & a == count of m]

m p a m p a m m
0 1 2 3 4 5 6 7

count a & p = 4

count of m = 4

ans = 1

Ques 2. Char Index

97 - 96

97 a → 1
98 b → 2
99 c → 3
d → 4
:
z → 26

a z d
↓ ↓ ↓
1 26 4

a 1 z 26 d 4

ans = " ";

```
char ch = Str.charAt(i)
int index = ch - 96;

ans = ans + ch + index;
```