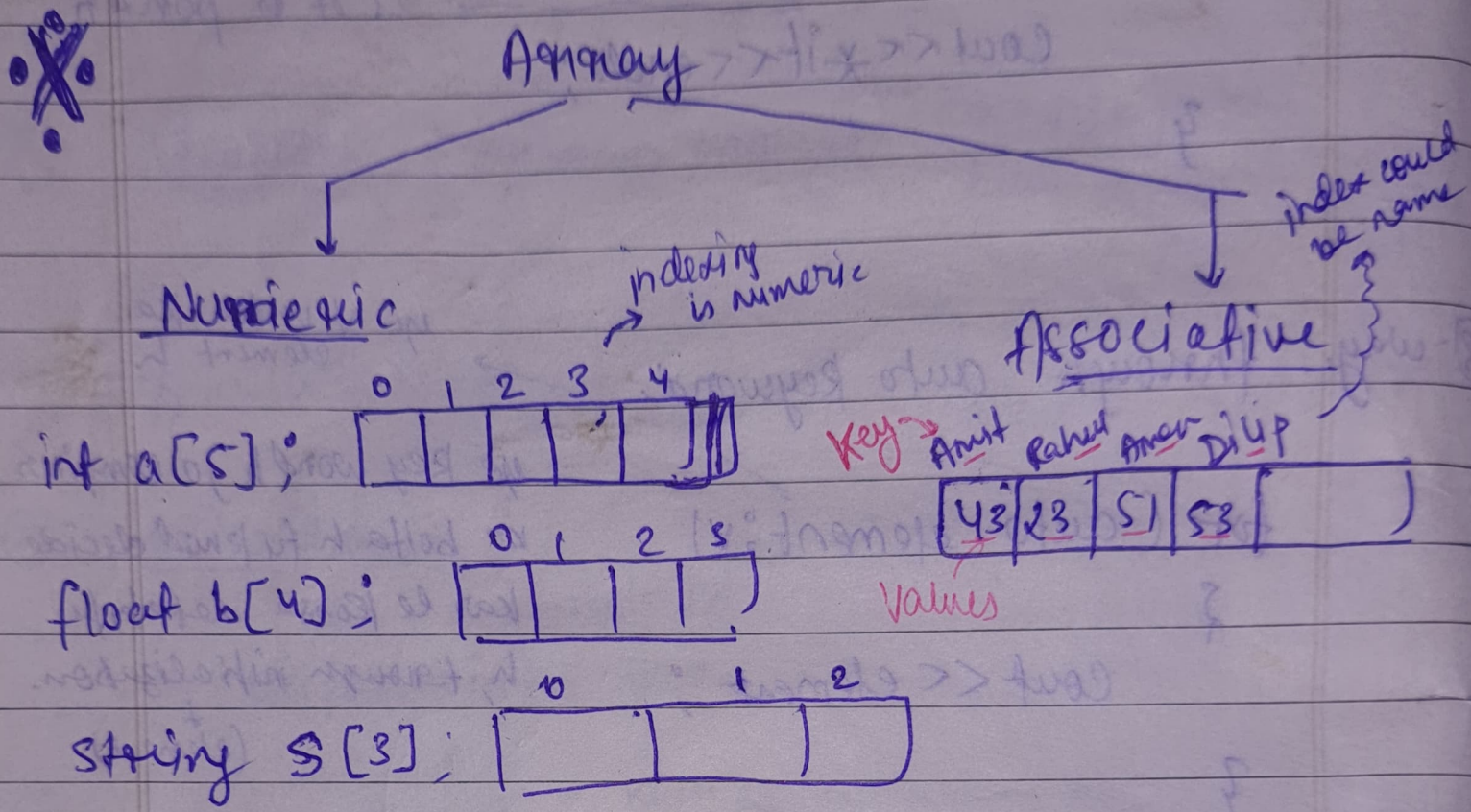
Array



- Maps are used to replicate associative arrays.
- Maps contain sorted key-value pairs, in which each key is unique and cannot be changed, and it can be inserted or deleted but cannot be altered.

→ Value associated with keys can be altered.

Page No.
Date :

→ maps always arrange its keys in sorted order.
→ In case the keys are ~~of~~ string type, they are sorted in dictionary order.

Q How to create a map ?

① way

→ key

```
map <int, string> customer;
```

~~key~~ → ~~key~~ value.

```
customer [100] = "Gajendra";
customer [123] = "Dilip";
customer [145] = "Adity";
customer [171] = "Shahid";
customer [200] = "Rajesh";
```

② nd way.

```
map <int, string> c{ {100, "Gajendra"},
  {123, "Dilip"}
};
```

```
→ customer.insert{ {5, "bheem"} };
→ customer.count (13);
→ customer.erase (5); → ~~value~~ key.
→ customer.find (100);
```

Page No.
Date :

Q. to access element

① way through (iterators)

```
map <int, string>::iterator p= customer.begin();

while ( p!=customer.end() ) {

    cout<<p->second<<endl;
    p++;
}.
```

→ we use first for key

① at   cout << customer.at (145);   → key] → print values of associated key

② size   customer.size() } → return size

③ Empty → return 0 if not empty
OR return 1 if empty

customer.empty ()

④ insert().

customer.insert{pair<int, string>{205, "saurabh"}}