(17) Object & wrapper Classes.

a) → Java has a top super class called obj class.

Has many methods, most popular ones are.

i) equals()
ii) ~~hashCode()~~ hashCode()
iii) toString()

i) equals() → == operator doesn't work when comparing objects based on values of member var. equals() is used.

ii) hashCode() & toString() convert obj data to hashCode & & iii) textual representation respectively.

b) wrapper class

→ Sometimes primitive data types are req to be converted to objects, to accomplish this Java has Boolean, Character, Integer, Long, Float & Double class. They are all called wrapper classes.

Belong to java.lang package

---

(18) final Keyword.

To make a variable, class (or) method immutable. (or) permanent.

final key comes after accen modifer & before return Type.

(21) Exception Handling

a) → when errors occur, to show more meaningful & user-friendly messages, handling of errors is done.

b) ⇒

try block → has code that might give error.

catch block → Specifies what is to be done when an exception occurs, therefore handling of exception occurs here.

c)
```
try{
  ---
}
catch(exception e){
  ---
}
```

e remains e only.

for every exception e, new catch block is needed, therefore for one try, multiple catch blocks might be needed.

d) In order to catch a new type of exception for which catch block is not there generic exception catch block can be used.. (is should be placed at the end.

e) Nested try - catch.

```
try {
    try {
    }
    catch (exception e) {
    }
}
catch (exception e) {
}
```

f) Suppose you want to display a 'thank you' message irrespective of whether there is exception or not.

then 'finally' is used..

```
finally {
}
```

g) Types of exceptions → Unchecked (Runtime exception)
→ Checked (all others).

h) Throwable
- Exception — Runtime or compile time errors
- Error — Seriously wrong, not to be caught.

9) Throw → User defined exceptions.

Syntax →

```
if (age[i] < 0) {
    throw new Exception ("Age -ve");
}
```

Also,

```
public void Fnname() throws Exception {
    // to be written inside which throw used.
}
```

Topics not covered now (Covered in OOPS later.)

1) Regex (Regular Expression)
2) Recursion
3) Package, protected & import.
   - Package → Inbuilt, User defined.
   - import → to use packages
4) Unit Testing & Code coverage
   - (JUnit & TestNG introduced).
   - softwares.
   - Code coverage → ensuring all code is tested.
5) Abstract and Interfaces.
