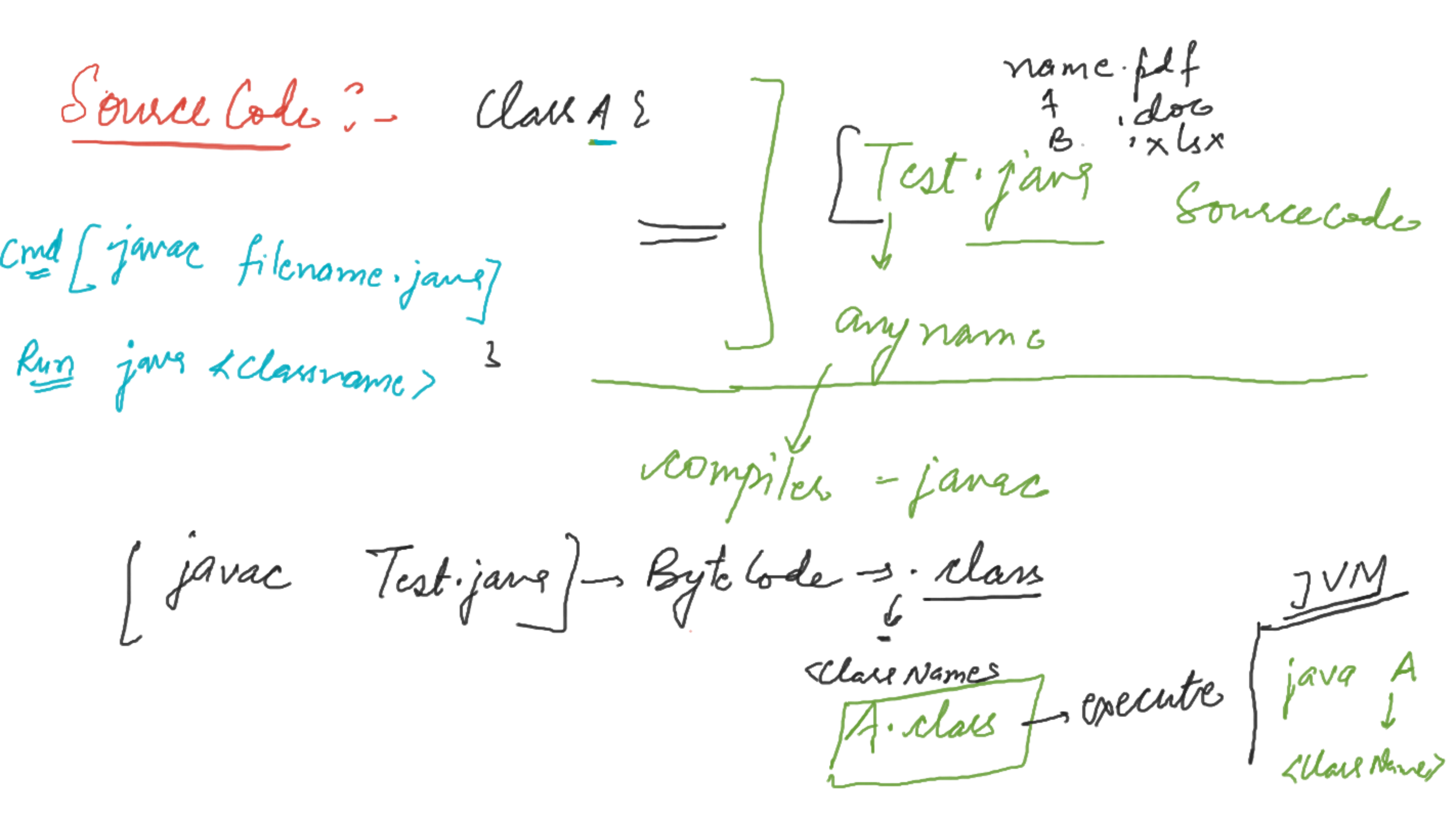
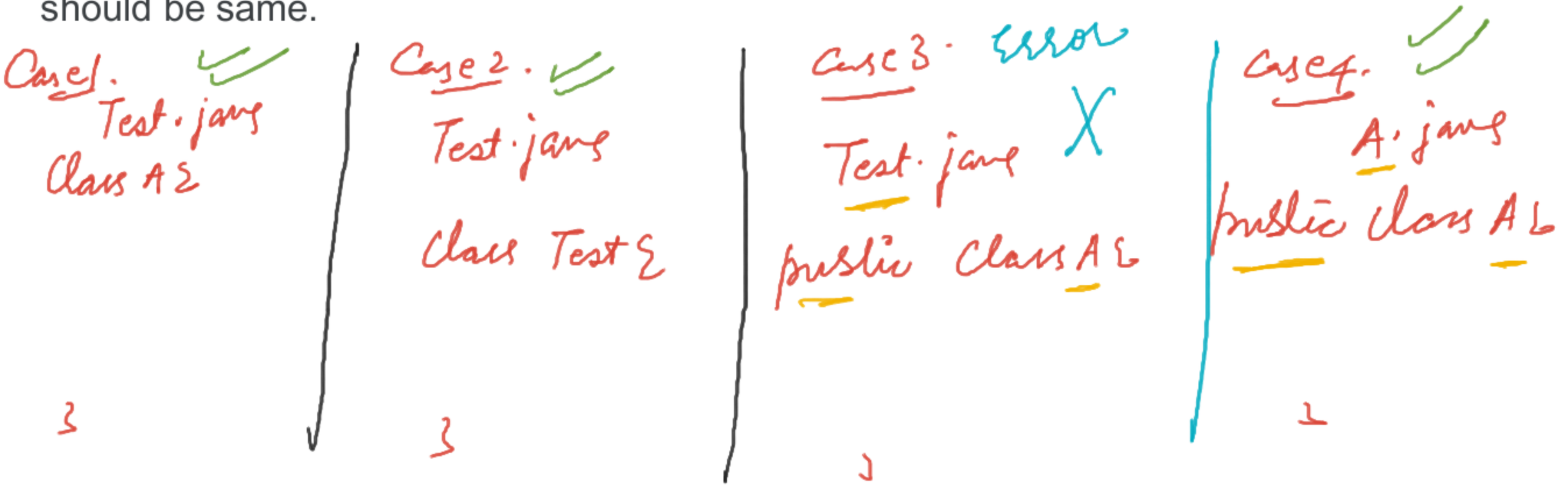

Source Code :-
class A {
}
name.pdf
A .doc
B .xlsx
Test.java
Source code
any name
cmd [javac filename.java]
Run java <classname>
compiler - javac
[javac Test.java] → Byte code → .class
JVM
<Class Names
A.class
execute
java A
<Class Name>

1. Class Name and FileName is different - Code will be compiled and run successfully.
2. ClassName and Filename is same - Code will be compiled and run successfully.
3. Class Name and FileName is different and using public keyword infront of className: Error

4. if we are using public keyword infront of className, then classname and FileName should be same.



# Name Convention

Class Name - always starts with uppercase - Alphabet, $

FileName — " " " - Alphabet

MethodName - " " " Lowercase - main()

Keyword - Builtin (Already) (Lowercase)

Variable - Lowercase, args

Can't start with number

$, _, letter, char

Keywords are reserved word used for special purpose - if, else, for, do while, while

- Keyword (predefined)
- Identifier ← className, varname, methodname
- Literal - any value
  - predefined Literals [null, true, false]

```
public class Test{
    public static void main(String[] args){
        System.out.print("Hello Java Code.....");
    }
}
```

k, I(c), I(m), I(c), I(v), I(c), I(prop), I(m), (L)

# List of Java Keywords

| | | | | |
|---|---|---|---|---|
| boolean | byte | char | double | float |
| short | void | int | long | while |
| for | do | switch | break | continue |
| case | default | if | else | try |
| catch | finally | class | abstract | extends |
| final | import | new | instance of | private |
| interface | native | public | package | implements |
| protected | return | static | super | synchronized |
| this | throw | throws | transient | volatile |

Variable → Holder, Container which hold/ Store the value

```
int a = 10;
boolean b = true;
char c = 'a';
String d = "Test";
```

'1' → char

```
$a = 20;
```

Date - 13/9

(1) Source Code → class A { . . . } → Save → Test.java

(2) Compile - javac Test.java → ByteCode → A.class

ext: .class

ClassName: A

(3) Interpret / Execute / run → java A

javac <filename>.java
java <classname>

```
int a = 10;   // variable Declaration and initialisation
int b;        // variable decl
b = 20;       // only initialisation / value assign
int b = 100;  X  if variable is already declared
                 and creating a new one with same name
                 then will throw compilation error.

int a, b, c;        // variable decl.
int a=10, b=20, c=30;  // variable decl.
```

int a=10;

System.out.println("value of a is 10"); //

int a=10;

```
System.out.println(a);//var value
System.out.println("a");
System.out.println("abvcsfsfd");
```

o/p

```
E:\students_data\Jagroop_QA\LearnJava>javac PrintVals.java

E:\students_data\Jagroop_QA\LearnJava>java PrintVals
10
a
abvcsfsfd

E:\students_data\Jagroop_QA\LearnJava>
```