Java Basics to Web Service

- Basics } 4
- OOP }
- web service (Jax-RS, MapStruct) → 2
- Testing (Mockito) → 1

Day 1

- selection (if, switch)
- Loops (for, while, do while)
- Strings
- Arrays

```
if (bool-expr) {
    ____
    ____
} else if (bool-expr) {
    ____
    ____
} else if (bool-expr) {
    ____
    ____
}
.
.
.
else {
    ____
    ____
}
```

T

F, T

all false

Bool Expr

>, <, >=, <=

==, !=

Logical operators

bool-expr && bool-expr

~ || ~

! bool-expr

```
switch (expr) {
    case value1:
        ____
        ____
        break;
    case value2:
        ____
        ____
        break;
    .
    .
    .
    default:
        ____
        ____
}
```

```
switch (expr) {
    case value1 -> ____;
    case value2 -> ____;
    .
    .
    .
    default -> ____;
}
```

```
x = bool-expr ? value if true : value if false
```

day

1 - 5 ⇒ work-day
6, 7 ⇒ off-day
else ⇒ Invalid

day

7, 1-4 ⇒ work-day
5, 6 ⇒ off-day
else ⇒ invalid

number

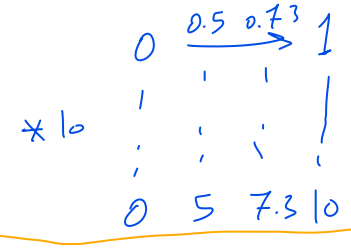
odd Even

what's 7 * 3?

>> 10 (wrong)

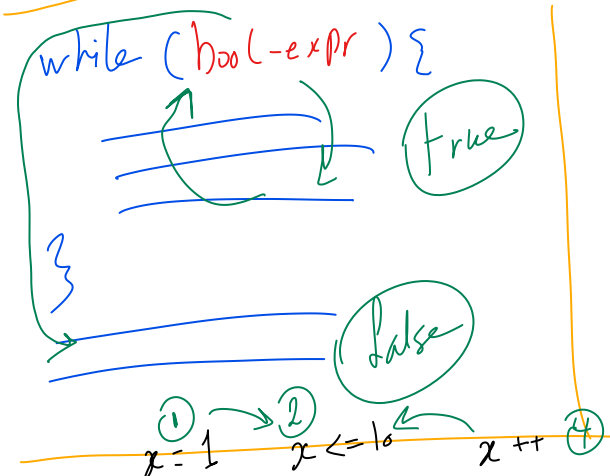
>> 21 (Correct)

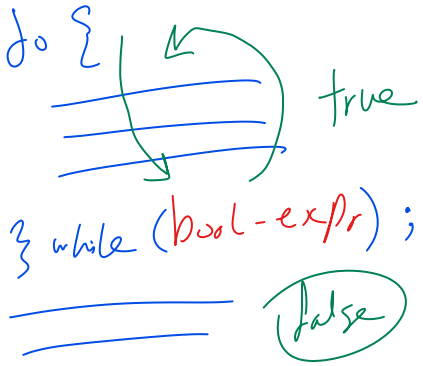
Math.random()

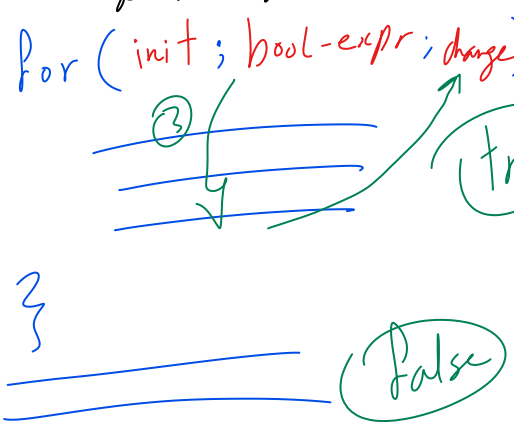


Math.floor()
Math.ceil()

int x = (int)7.3;







break

continue

what's 3 * 5?

>> 15

what's 2 * 4?

>> 8

what's 7 * 10?

>>

>> 13

Game over!

what's 7 * 3?

⇒ 10

Try again, what's 7*3?

⇒ 10

Try again, what's 7*3?

⋮

⇒ 21

Correct

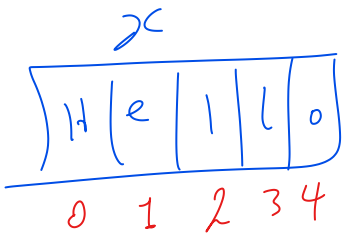
String x = "Hello";

sout(x) ⇒ Hello

sout(x.~~charAt~~(1)) ⇒ e

x.length() ⇒ 5

char g = x.charAt(1);



Character.

---

① at least 8 char → length

② at least 2 digits → isDigit

③ at least 2 upper → isUpper

2oAhmeD20

digitCount: ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

upperCount: ~~0~~ ~~1~~ 2

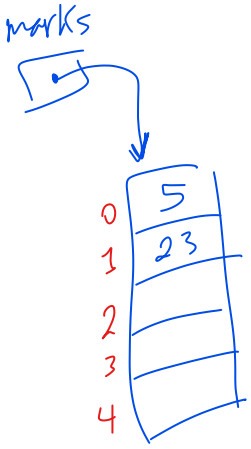
---

① exactly 10 char → length

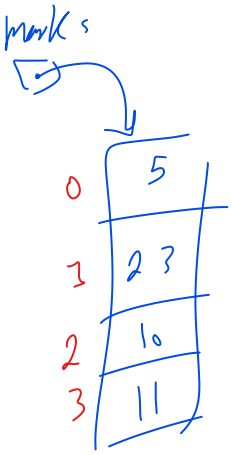
② starts with 05 → startsWith

③ all digits → isDigit

```
int[] marks = new int[5];
Sout(marks)      → @x....
Sout(marks[0])   → 0
marks[0] = 5;
marks[1] = 23;
```

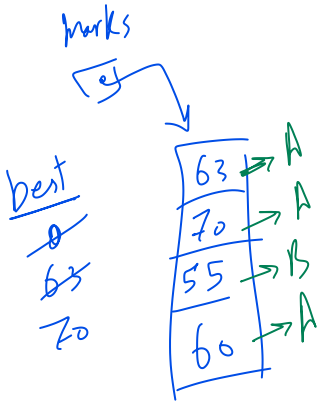



```
int[] marks = {5, 23, 10, 11}
```



Enter num of students: 4

Enter std mark: 63
: 70
: 55
: 60




```
m >= best-10 → A      (best 60)
m >= best-20 → B      (best 50)
m >= best-30 → C      (best 40)
m >= best-40 → D      (best 30)
else → F
```

best = 70

```
for (datatype var : collection) {
}
```