code decoded

Conditionals

boolean operators

and	&&	A && B
or	||	A || B
logical not	!	! B
equal to	==	A == B
not equal to	!==	A !== B

Boolean wrapper class

All primitive types have a wrapper class that stores the value of the primitive.

capital B

```
Boolean b = new Boolean(false);
```

Allows us to convert primitive ⟷ object

A primitive value cannot be null, but wrapper classes may be given a null value.

if-else statements

evaluated from top to bottom

When a condition evaluates to true, the block is run, and the rest of the ladder is skipped.

```
if (condition1) {
    // code goes here
} else if (condition2) {
    // code goes here
} else {
    // code goes here
}
```

The if keyword starts the conditional

an expression that evaluates to a boolean.

executes if the preceding condition(s) are false.

executes if none of the condition(s) evaluate to true.

switch statement

```
switch(expression)
  case value1:
    // code goes here
    break;
  case value2:
    // code goes here
    break;
  case value3:
    // code goes here
    break;
  default:
    // code goes here
```

If no match is found, the default code block runs.

ternary statement

: shorthand syntax to return a value based on the result of a condition.

```
(condition) ? return if true : return if false;
```

e.g.

```
boolean myBool = false;
int num = (myBool) ? 1 : 0;
```