Primitive Types

Integers

(`int`)

Lecture Contents

- List of Java Primitive Types
- Review of Number Systems
- How to Store a Binary Integer in a Computer
- Java `int` type

List of Java Primitive Types

- *Integers*
  - `byte` (8 bits)
  - `short` (16 bits)
  - **`int`** (32 bits) ← This Lecture
  - `long` (64 bits)
- *Real Numbers*
  - `float` (32 bits)
  - **`double`** (64 bits)
- *True/False*
  - **`boolean`** (1 bit?)
- *Letters*
  - `char` (16 bits)

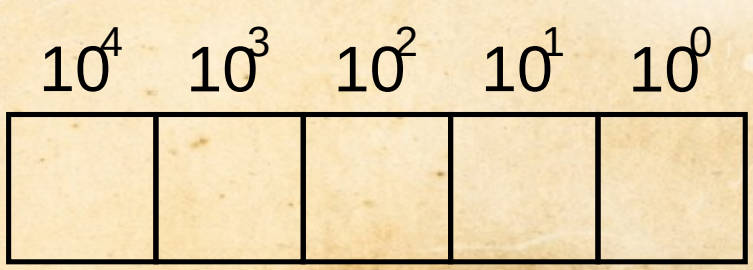
Note: Primitive types tested in *AP Computer Science A* are given in bold font.

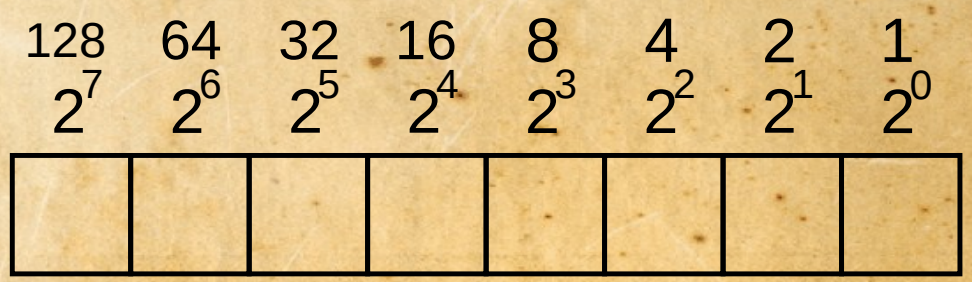
Review of Number Systems

- The ***denary*** (a.k.a. ***decimal***) system uses ***place value***, ten digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), and a decimal point to represent a number.

Review of Number Systems

- The ***binary*** system uses ***place value*** and two digits (0, 1) to represent a value.
- Although we could, we generally don't see a decimal point used with binary numbers.
- For denary, each place value is a power of ten. For binary, each place value is a power of two.

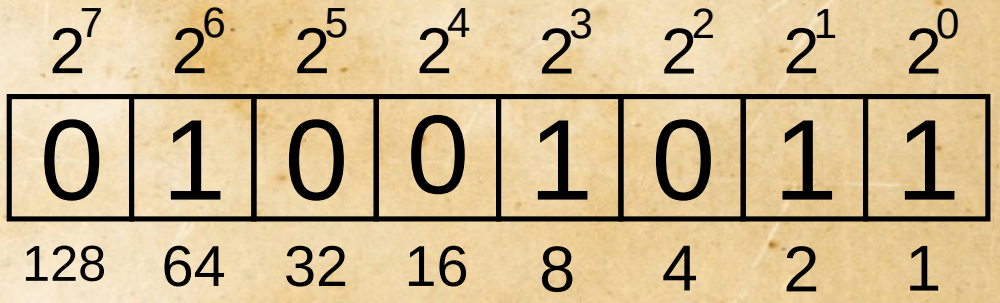




Review of Number Systems

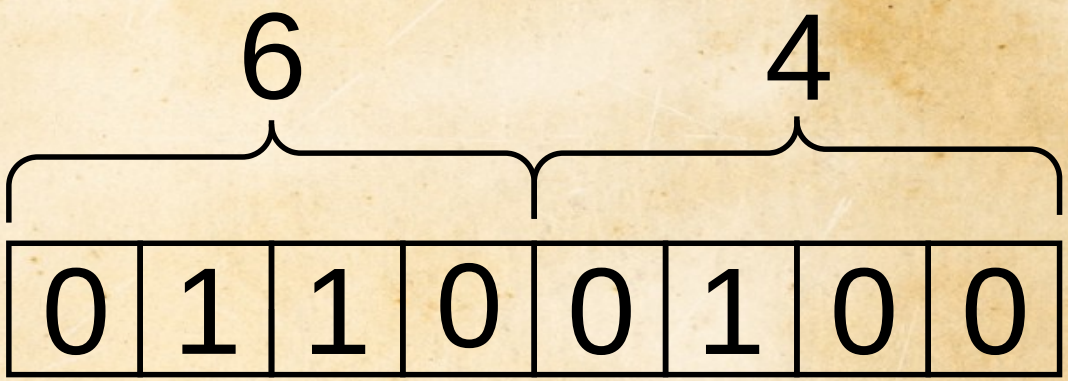
- Be able to convert back and forth between ***binary*** and ***denary***.

75−64 = 11
11−8 = 3
3−2 = 1
1−1 = 0



Review of Number Systems

- Be able to convert back and forth between ***binary*** and ***hexadecimal***.



Decimal	Hexa-decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Lecture Contents

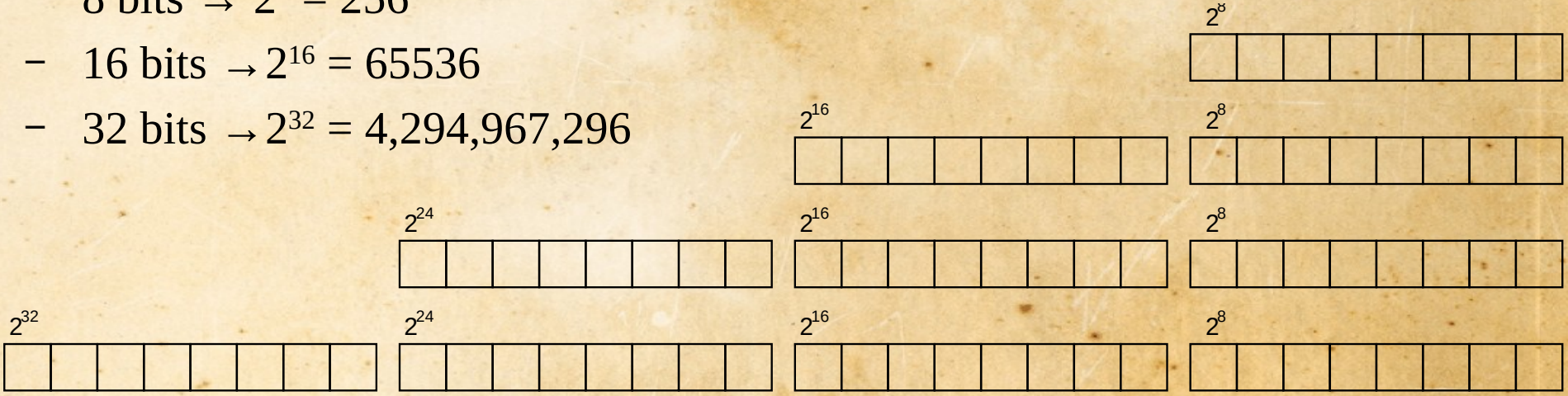
- Review of Number Systems
- **How to Store a Binary Integer in a Computer**
- Java `int` type

How to Store a Binary Integer in a Computer

- Modern computers organize ***bits*** into ***bytes*** (8 bits = 1 byte).
- Computer hardware is designed to operate on a set number of bytes.
  - The more bytes we operate on at once, the more complex the hardware must be.
- How many bytes should we use to store each integer?

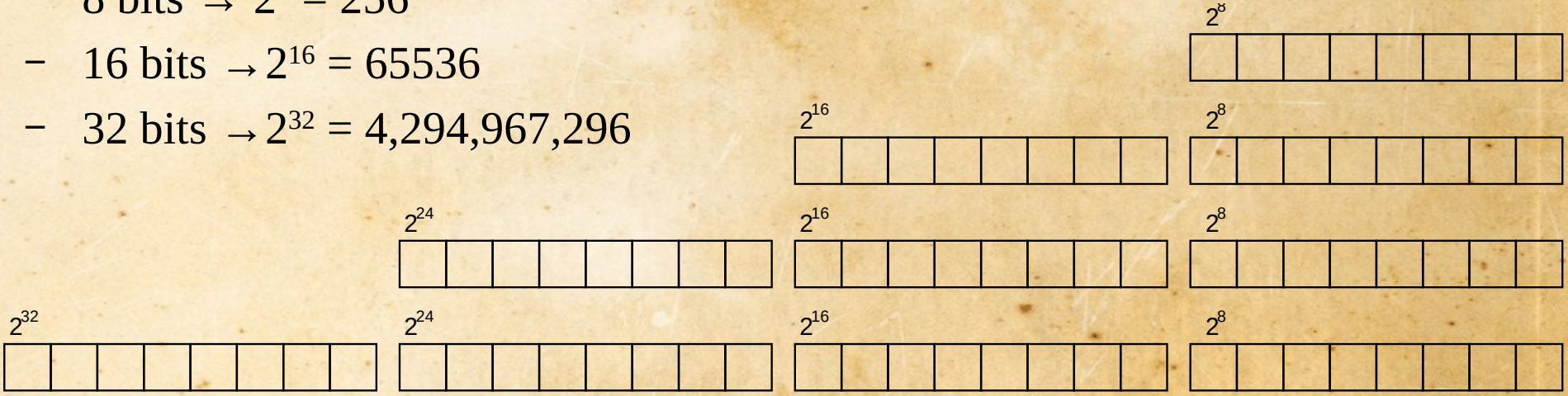
How to Store a Binary Integer in a Computer

- Modern computers organize ***bits*** into ***bytes*** (8 bits = 1 byte).
- Computer hardware is designed to operate on a set number of bytes
- How many bytes should we use to store each integer?
  - 8 bits → $2^8 = 256$
  - 16 bits → $2^{16} = 65536$
  - 32 bits → $2^{32} = 4{,}294{,}967{,}296$



How to Store a Binary Integer in a Computer

- Modern computers organize ***bits*** into ***bytes*** (8 bits = 1 byte).
- Computer hardware is designed to operate on a set number of bytes
- How many bytes should we use to store each integer?
  - 8 bits → $2^8 = 256$
  - 16 bits → $2^{16} = 65536$
  - 32 bits → $2^{32} = 4{,}294{,}967{,}296$



How to Store a Binary Integer in a Computer

- Java programmers can choose from the following ***types*** for storing integers

type	bits	numerical range
byte	8	-128 to +127
short	16	-32768 to +32767
int	32	-2,147,483,648 to +2,147,483,647
long	64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

  - For this course, unless you have a specific reason, always use the type **`int`** to store integers.

Lecture Contents

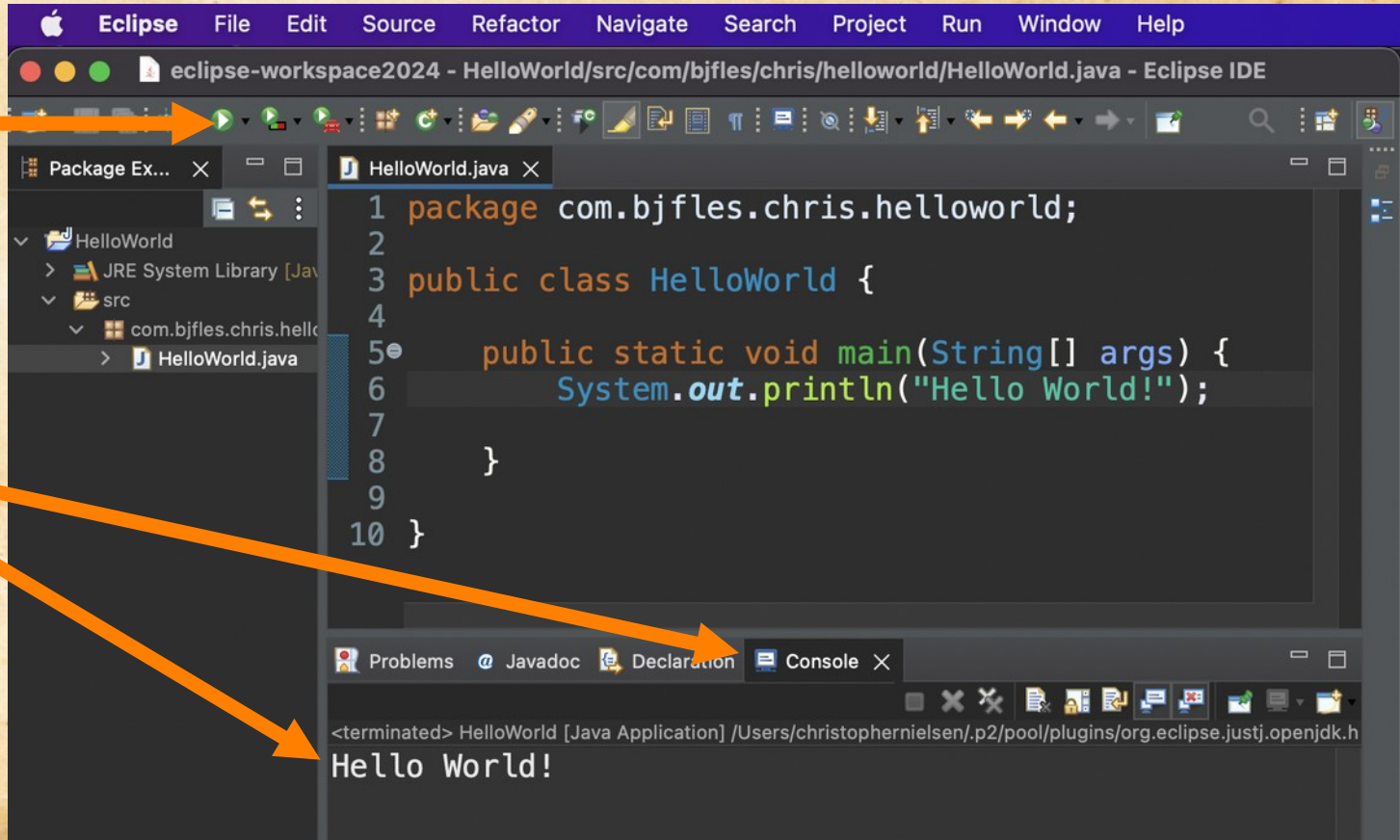
- Review of Number Systems
- How to Store a Binary Integer in a Computer
- **Java `int` type**

Java `int` Type

Ensure you were able to get the "`HelloWorld`" assignment completed.

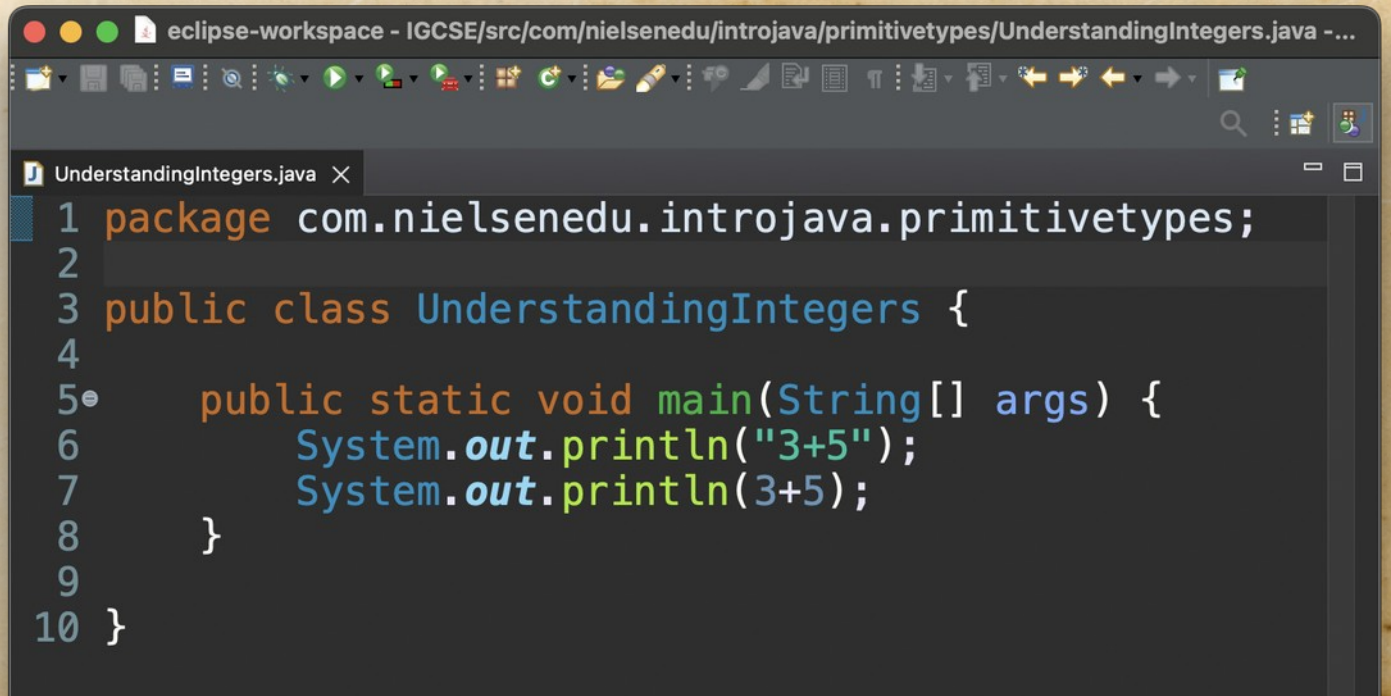
Click here to run.

Check the output in the console.



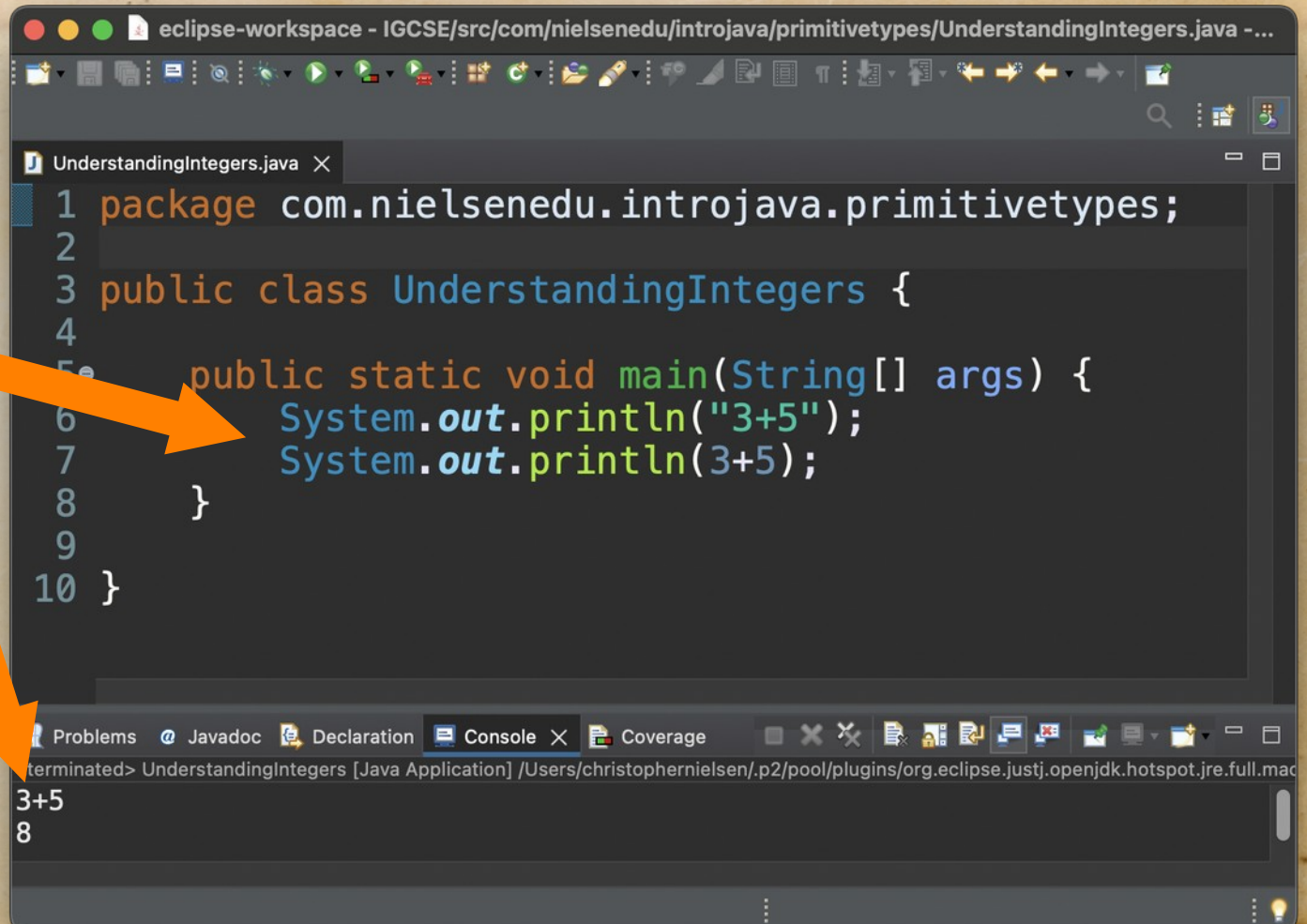
Java `int` Type

- Create a new class named "`UnderstandingIntegers`"
  - What output do you expect from the program, below?



Java `int` Type

Compare



Java `int` Type

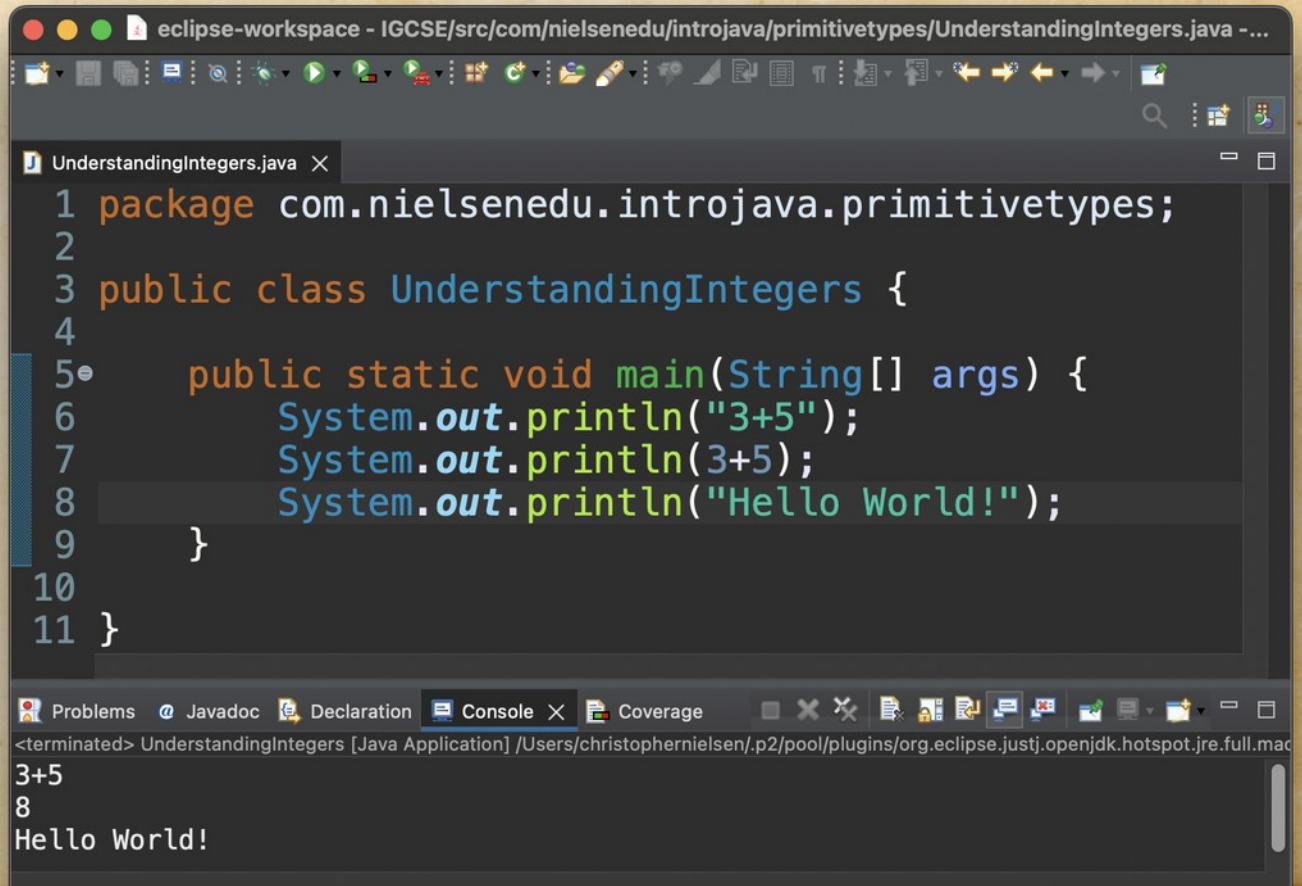
- Anything enclosed double quotation marks (") is NOT considered a number by Java, it is considered a `String`.
  - We will learn more about the `String` type later.
  - `System.out.println("3 + 5")` will print exactly what is found within the quotation marks: `3 + 5`.

Java `int` Type

- Integers that are not enclosed double quotation marks (") are considered ***integer literals***.
  - Operations, such as addition, can be performed on these integers.
  - `System.out.println(3 + 5)` will perform the addition calculation, then print the result: `8`.

Java `int` Type

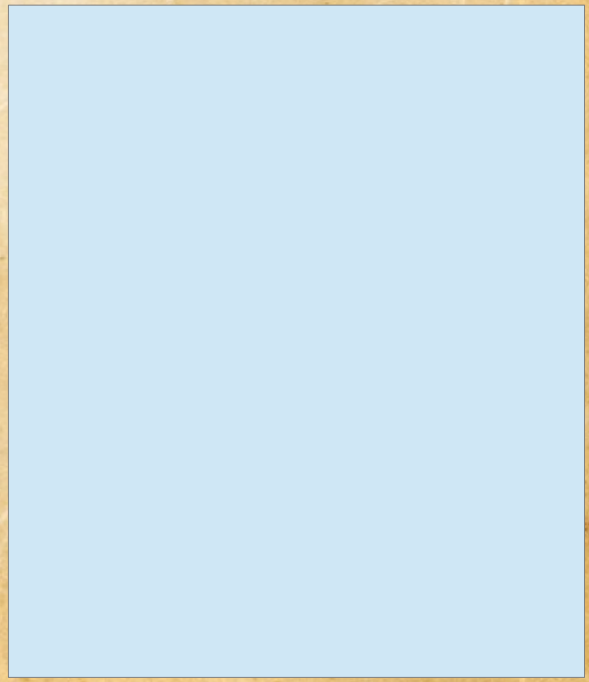
- Now we'll add more lines to our `UnderstandingIntegers` class...



Java `int` Type

- Add each line to your `UnderstandingInteger` class and observe the output.

```
System.out.println("Hello World!");

System.out.println("Hello " + "World!");

System.out.println("Hello " + World);

System.out.println("3 + 5");

System.out.println("3" + "5");

System.out.println(3 + 5);
```



Java `int` Type

- Add each line to your `UnderstandingInteger` class and observe the output.

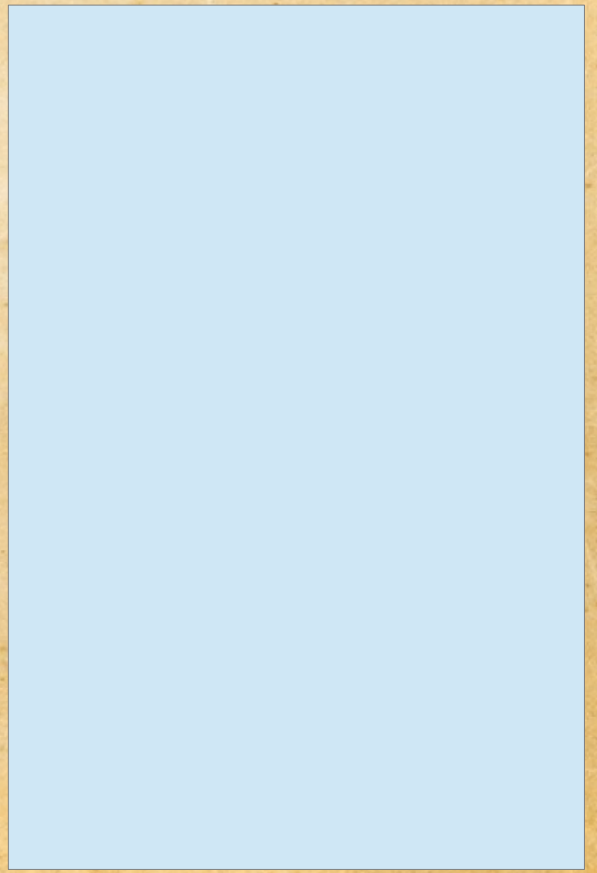
```
System.out.println("Hello World!");
System.out.println("Hello " + "World!");
System.out.println("Hello " + World);
System.out.println("3 + 5");
System.out.println("3" + "5");
System.out.println(3 + 5);
```

```
Hello World!
Hello World!
error
3 + 5
35
8
```

The + operator will *concatenate* strings and add numbers.

Java **`int`** Type

```
System.out.println("Hello " + 3);

System.out.println("Hello " + 3 + 5);

System.out.println("Hello " + (3 + 5) );

System.out.println(3 + 5 + " Hello");

System.out.println(3 + (5 + " Hello") );
```



Java `int` Type

```
System.out.println("Hello " + 3);
System.out.println("Hello " + 3 + 5);
System.out.println("Hello " + (3 + 5) );
System.out.println(3 + 5 + " Hello");
System.out.println(3 + (5 + " Hello") );
```

```
Hello 3
Hello 35
Hello 8
8 Hello
35 Hello
```

- **When a string and integer (`int`) are operands to a + operation, the integer is converted to a character string and concatenated to the string.**
- **The + operations are performed from left to right.**

Java `int` Type

- The + operator will *concatenate* strings and add numbers.
  - `System.out.println("Hello " + "World!");` → `Hello World!`
  - `System.out.println(3 + 5);` → `8`
- When a string and integer (`int`) are operands to a `+` operation, the integer is first converted to a character string, and then concatenated to the string.
  - `System.out.println("Hello " + 3);` → `Hello 3`
- The `+` operations are performed from left to right.
  - `System.out.println("Hello " + 3 + 5);` → `Hello 35`

List of Java Primitive Types

- *Integers*
  - `byte` (8 bits)
  - `short` (16 bits)
  - **`int`** (32 bits) ← This Lecture
  - `long` (64 bits)
- *Real Numbers*
  - `float` (32 bits)
  - **`double`** (64 bits)
- *True/False*
  - **`boolean`** (1 bit?)
- *Letters*
  - `char` (16 bits)

Note: Primitive types tested in *AP Computer Science A* are given in bold font.

Primitive Types

Integers

(`int`)