

Lec 4

Variables and math

Announcements

- If you did not get an email from your tutor, message Head Tutor, Jason Wong, jwwong@ucsd.edu
- HW2 posted
- Quizzes and midterm moved to Fridays

Variables

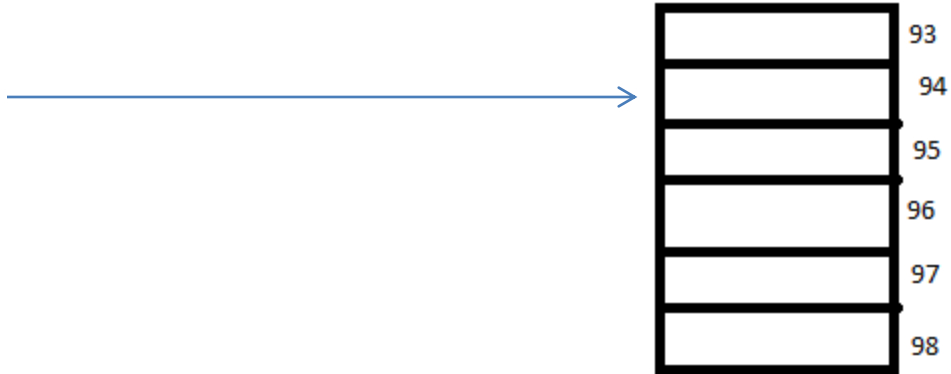
- What is the output of this code?

```
int x = 3;  
int y = x;  
System.out.println(y == 3);  
x = 4;  
System.out.println(y == 3);
```

- ➡ A) true B) false C) true D) false E) None of the
 true false false true above

Variables

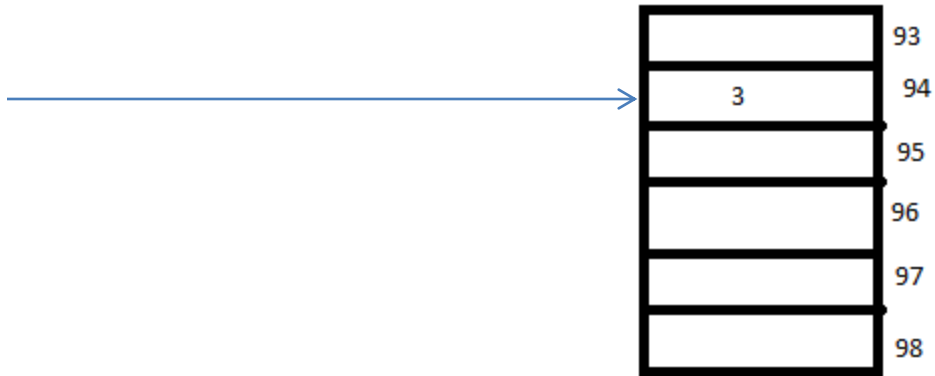
`int x;`



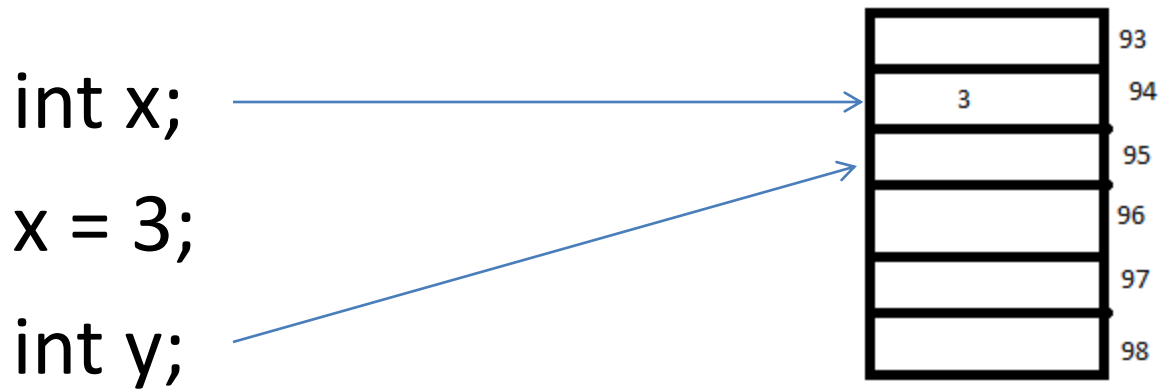
Variables

```
int x;
```

```
x = 3;
```



Variables



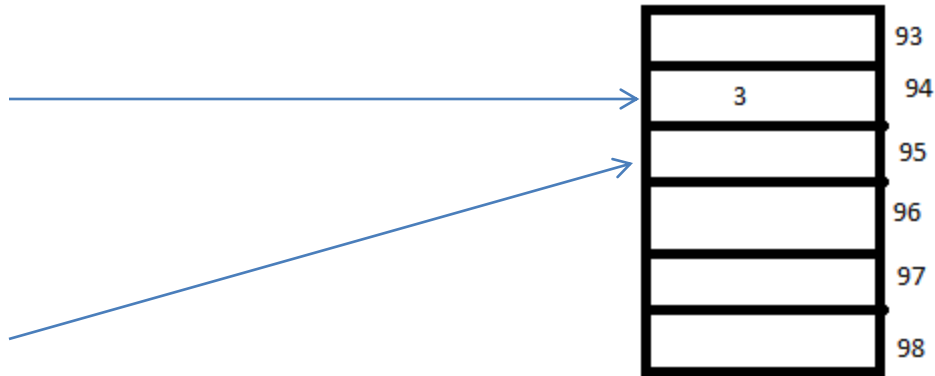
Variables

int x;

x = 3;

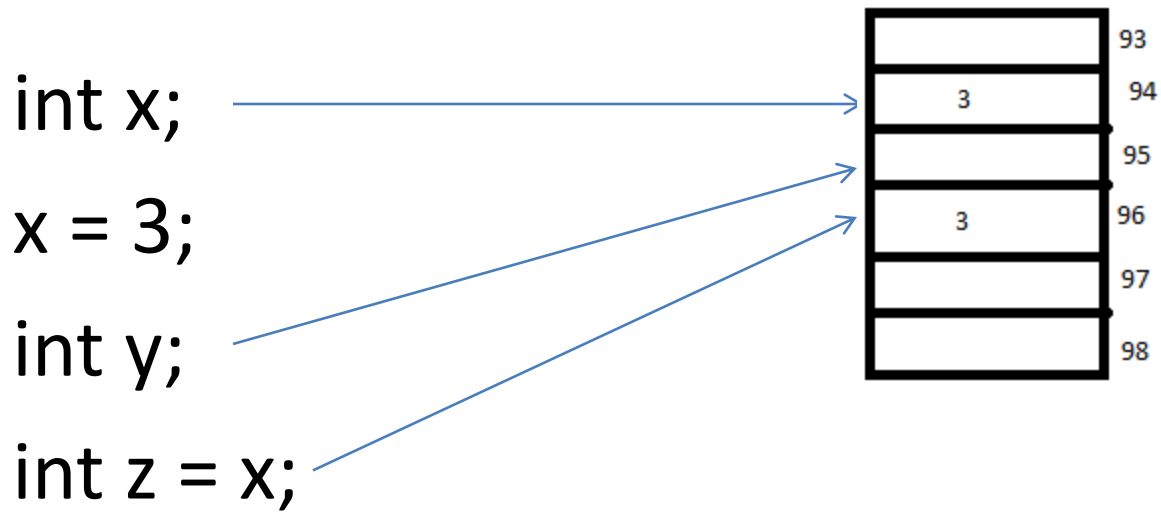
int y;

int z = x;



?

Variables



Variables

- What is the output of this code?

```
int x = 3;  
int y = 0;  
x = y;  
y = x;  
System.out.println(y == 3);  
x = 3;  
System.out.println(y == 3);
```

- A) true  B) false C) true D) false E) None of the
true false false true above

Declaring Variables

```
Type variableName = value;
```

- Primitive types:
 - int, long, short – standard integers, e.g. 5, 100, -37
 - double, float – decimal numbers, e.g. 5.32, -100.2
 - char – single character, e.g. 'C', '!'
 - boolean – true/false
 - boolean flag = true;

Naming variables

- Must start with a letter (upper or lower case)
- Can have digits (0-9)
- _ and \$ are considered letters
- Cannot be a reserved word
 - E.g. float, public, short

Which of the following contains all valid variable names?

- ➡ A) pizza_party, _pepperoni_, chee\$y
- B) Round.Table., fiveToppings, 1olive
- C) Public, pizza_party, int
- D) All of the above
- E) None of the above

Naming Variables

- Case sensitive
- Start with lower case, if multiple words, make next word capital
 - `int numStudents;`
- Be descriptive, but not ridiculous
 - `thisIsTheValueOfA`

Overflow

- 32 bits to represent an int
 - 1 bit for sign
 - 31 bits for value
 - Max value of 2,147,483,647

Integer division

What is the result of the following:

```
int x = 5 / 2;
```

x = ?



A) 2

B) 3

C) 2.5

D) Compiler error

E) Runtime error

Integer Division and modulo

`int x = 5 / 2;` $\rightarrow 2 \frac{1}{2}$

Integer division takes the quotient (2)

`int x = 5 % 2;` $\rightarrow 2 \frac{1}{2}$

Modulo takes the remainder (1)

Examples

$$7 \% 9 = 7$$

$$10 \% 2 = 0$$

$$0 \% 5 = 0$$

```
public class DivError {  
  
    public static void division(int x, int y) {  
        x = x / y;  
    }  
  
    public static void main(String[] args) {  
        int i = 10;  
        int b = 0;  
        division(i, b);  
    }  
}
```

You type the following and get:

\$ java DivError

Exception in thread "main" java.lang.ArithmeticException: / by zero
 at DivError.division(DivError.java:4)
 at DivError.main(DivError.java:10)

Was this a (1) compile time error or runtime error, (2) on what line of code is the problem and (3) what is the reported error?

A) Compiler error. Can't tell the line. Bad code

B) Runtime error. Line 4. / by zero

C) Runtime error. Line 10. java.lang.ArithmeticException:

D) Compiler error. Line 4. / by zero

E) Compiler error. Line 10. DivError.java

Order of Operations

- Just like regular math:

```
int x = 4;  
int y = 3 + x / (2 * -1);  
    //y = 3 + 4 / (2 * -1);  
    //y = 3 + 4 / -2;  
    //y = 3 - 2;  
    //y = 1;
```

To understand how the program works, only do 1 step at a time

Expressions

- Just like regular math:

()

pre ++ -- (e.g. ++var, --var)

Type cast

* / %

+ -

< <= > >=

== !=

&& ||

=

post ++ -- (e.g. var++, var--)

Compound operators

- Different ways to write the same thing

```
num = 3;
```

```
num = num - 1;
```

```
num -= 1;
```

```
num--;
```

```
--num;
```

Example

What gets printed:

```
int j = 10;  
System.out.println( j++);  
j = 10;  
System.out.println( ++j);
```

A) 10 B) 11 C) 10
10 11  11

D) ++j E) None of the
j++ above

()

pre ++ -- (e.g. ++var, --var)

Type cast

* / %

+ -

< <= > >=

== !=

&& ||

=

post ++ -- (e.g. var++, var--)

Programmers' code

- If it's not 100% clear, use ()
- Never use ++ inside of expressions