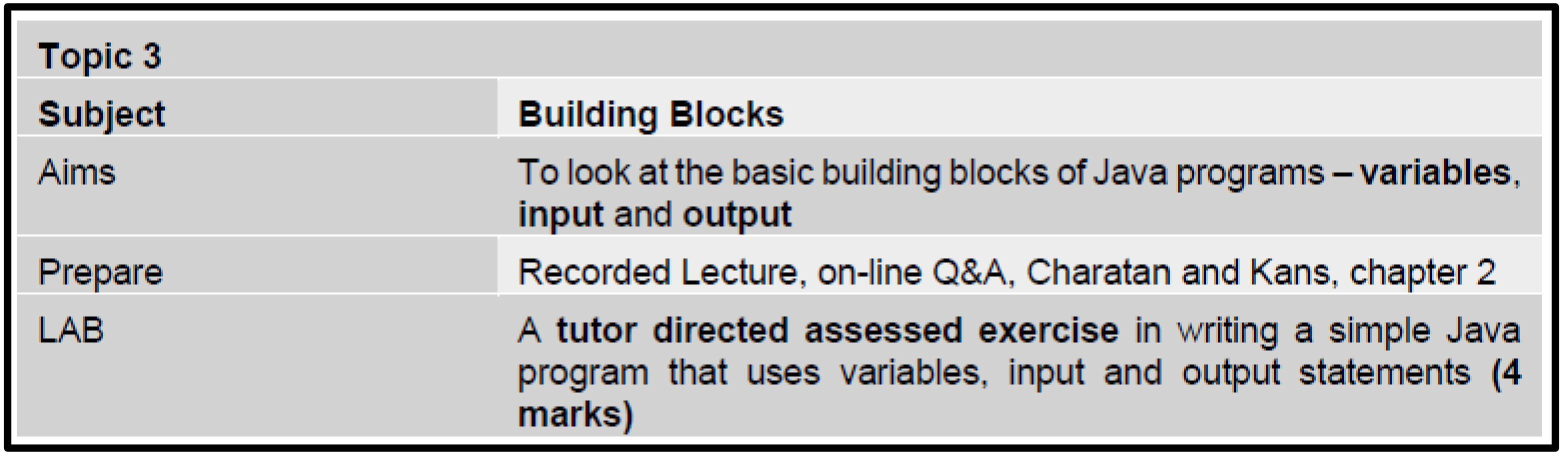
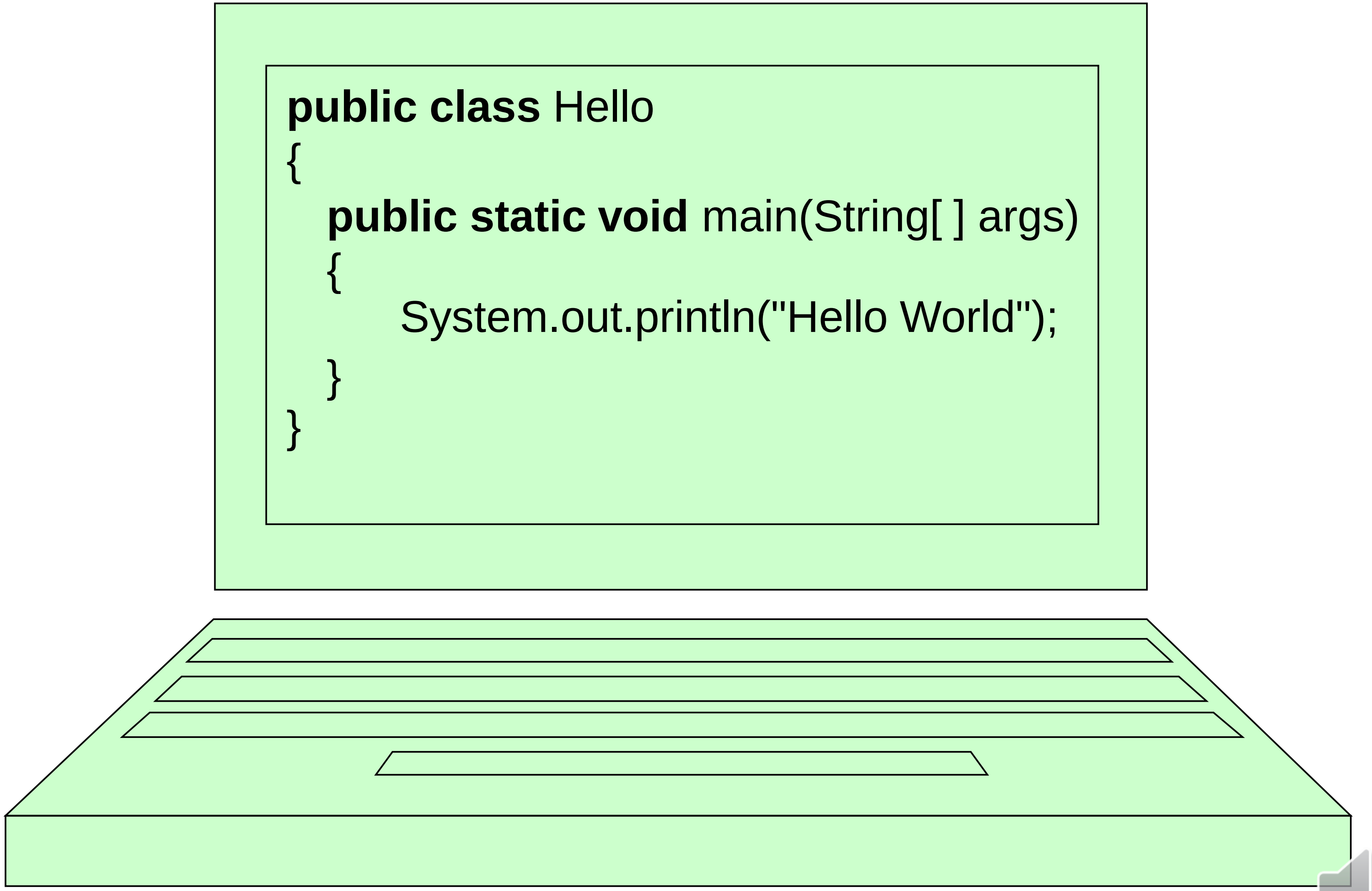
CN4001/ CD4001
Software Development
Topic 3: Building Blocks
Part 1

Module Study Guide

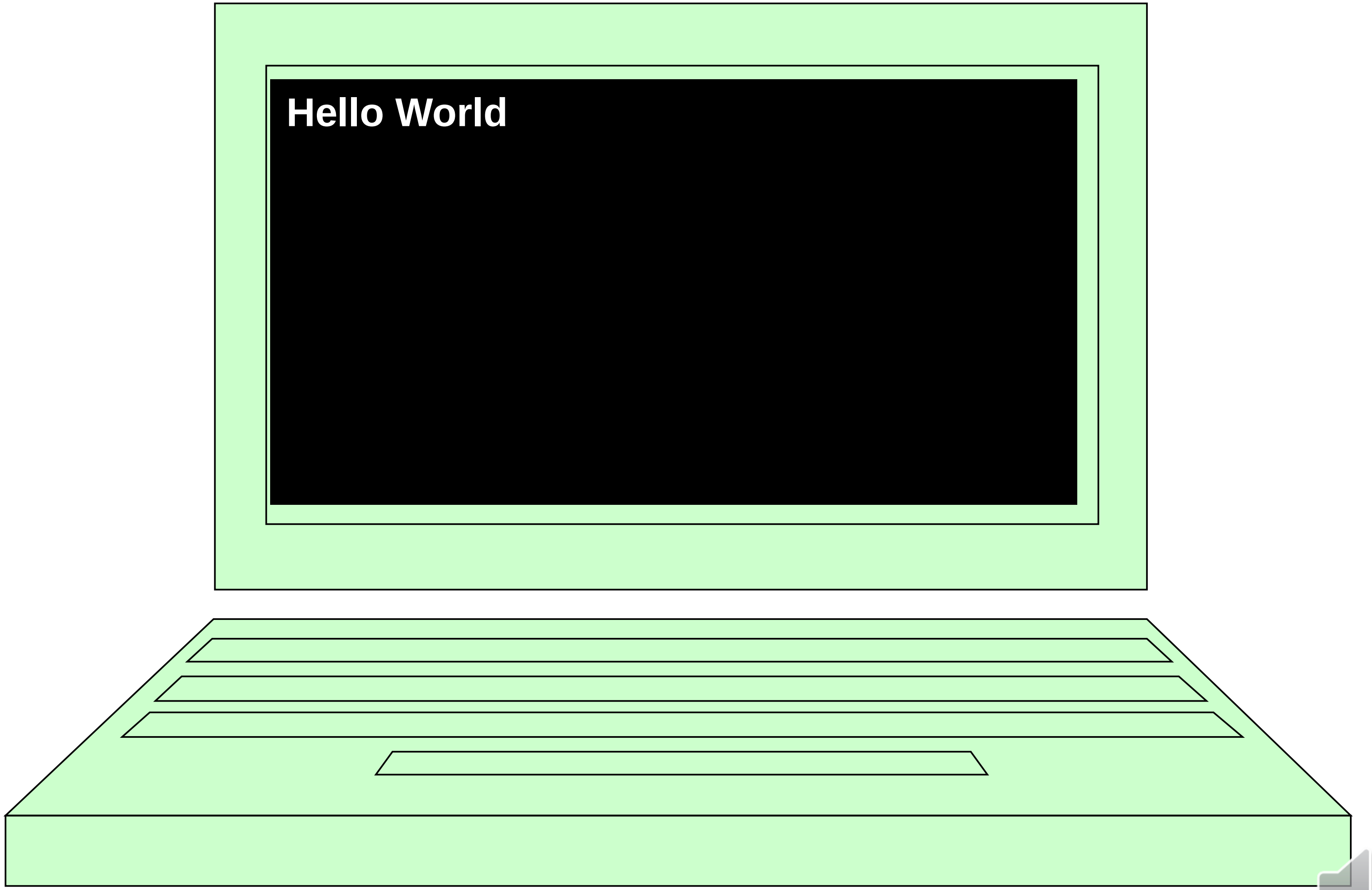


Topic 3	
Subject	**Building Blocks**
Aims	To look at the basic building blocks of Java programs – **variables**, **input** and **output**
Prepare	Recorded Lecture, on-line Q&A, Charatan and Kans, chapter 2
LAB	A **tutor directed assessed exercise** in writing a simple Java program that uses variables, input and output statements **(4 marks)**

Hello World!

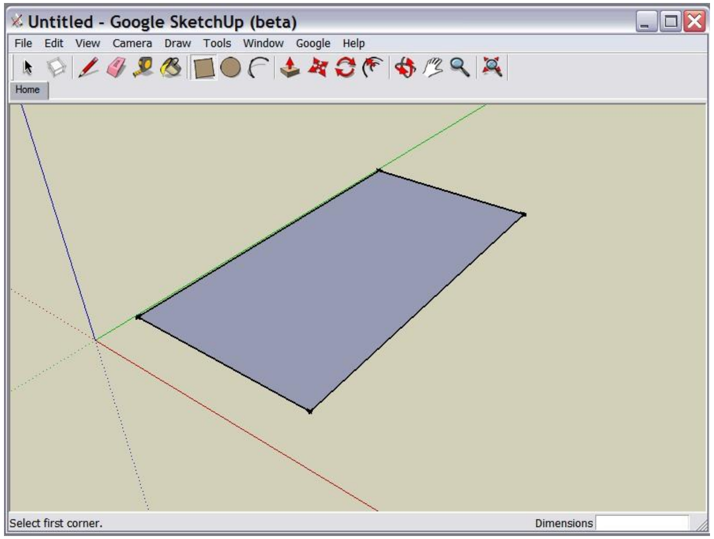


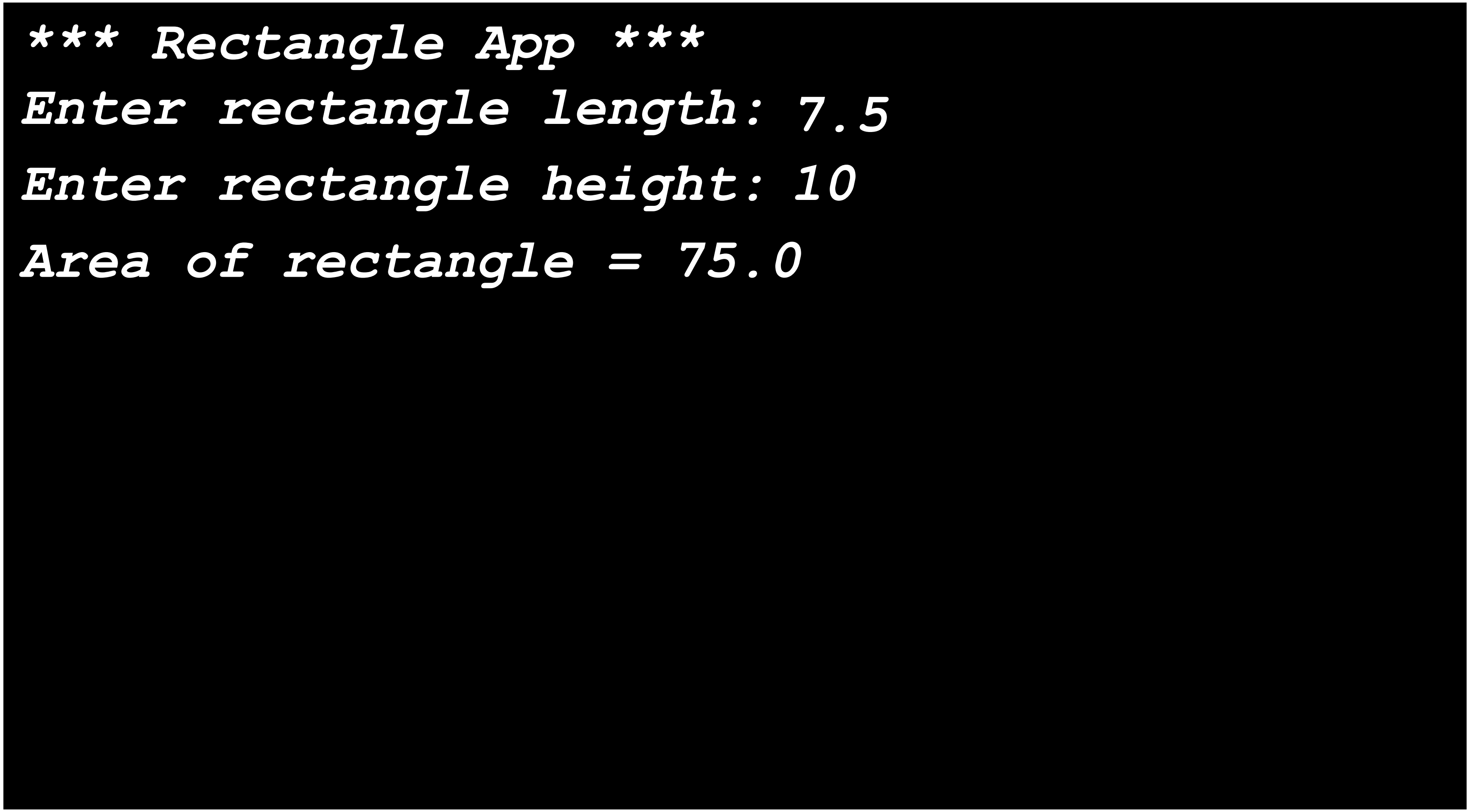
```
public class Hello
{
   public static void main(String[ ] args)
   {
          System.out.println("Hello World");
   }
}
```



```
Hello World
```

Developing a Java Application





```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

RUN

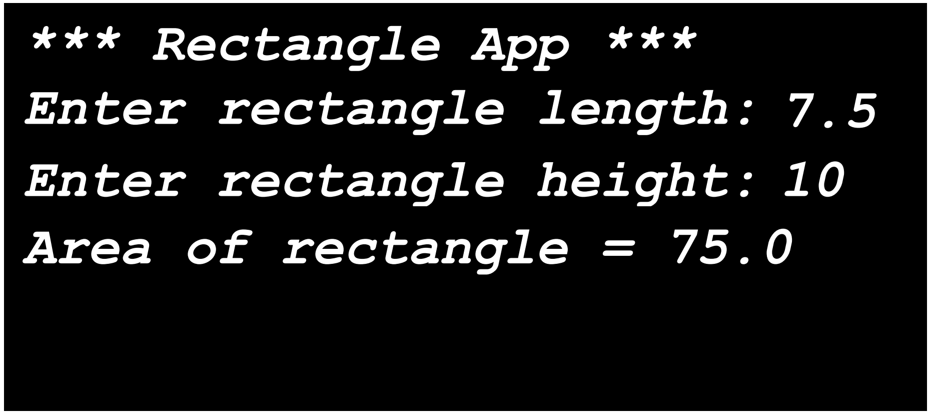
Activity

“How many instructions have to be executed in this interaction?”

```
*** Rectangle App ***

Enter rectangle length: 7.5


Enter rectangle height: 10



Area of rectangle = 75.0
```

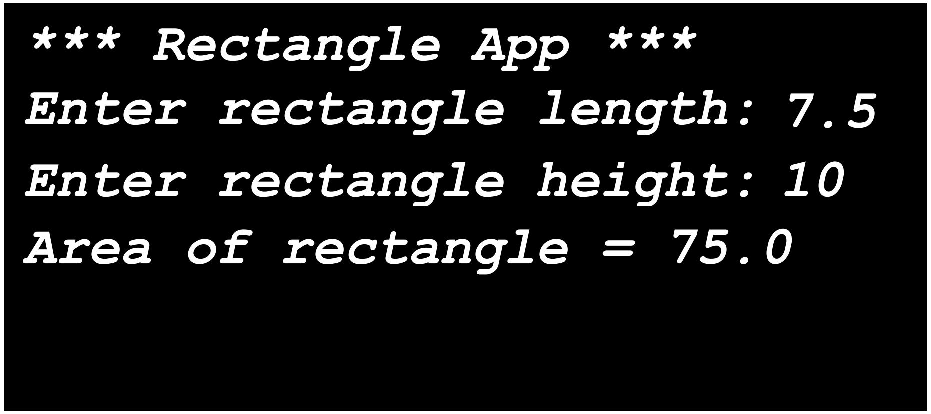
```
DISPLAY title

DISPLAY prompt for length
ENTER length

DISPLAY prompt for height
ENTER height

DISPLAY area
```

Let’s try and write this program together



```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

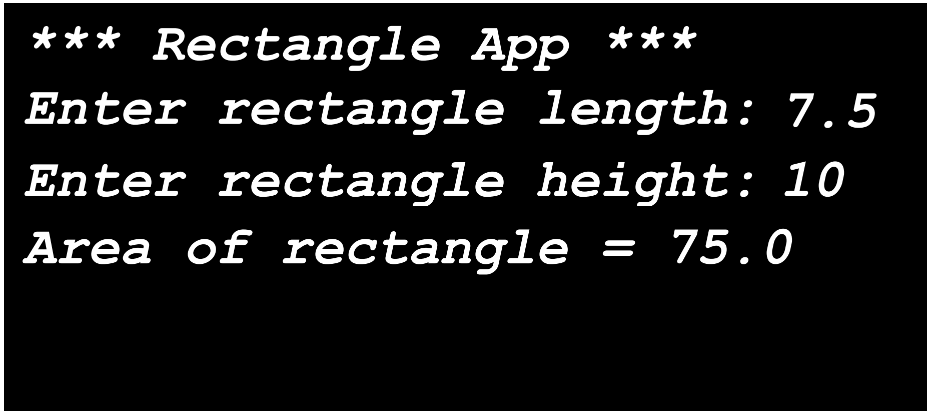
```
DISPLAY title

DISPLAY prompt for length
ENTER length

DISPLAY prompt for height
ENTER height

DISPLAY area
```



```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

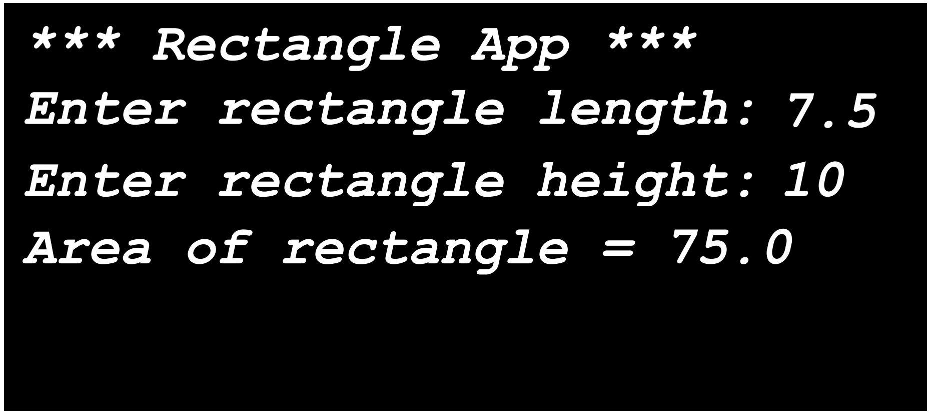
```
System.out.println(“*** Rectangle App ***”);
```

```
DISPLAY prompt for length
ENTER length

DISPLAY prompt for height
ENTER height

DISPLAY area
```



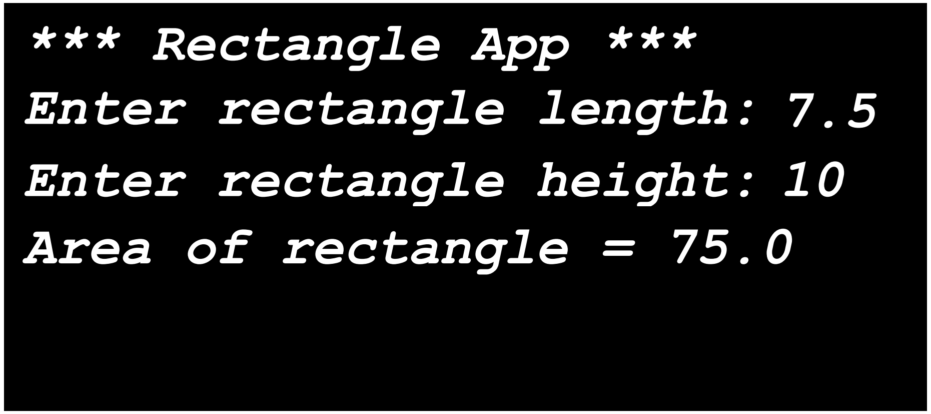
```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

```
System.out.println(“*** Rectangle App ***”);
System.out.print(“Enter rectangle length: ”);
ENTER length
DISPLAY prompt for height
ENTER height
DISPLAY area
```



```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

```
System.out.println(“*** Rectangle App ***”);
System.out.print(“Enter rectangle length: ”);
    ENTER length
System.out.print(“Enter rectangle height: ”);
    ENTER height
    DISPLAY area
```



```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

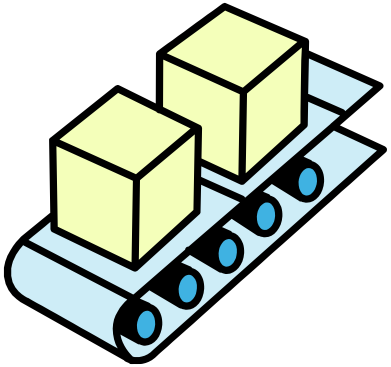
```
System.out.println(“*** Rectangle App ***”);
System.out.print(“Enter rectangle length: ”);
    ENTER length
System.out.print(“Enter rectangle height: ”);
    ENTER height
System.out.println(“Area of rectangle = ” ?? );
```

Variables in Java

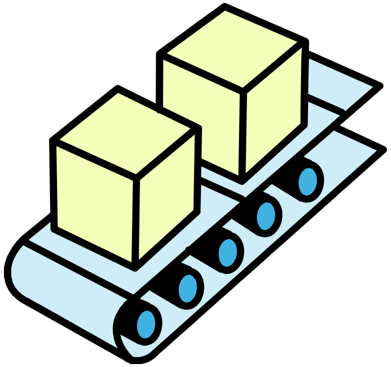
Variables in Java

Variables are named locations in memory that can store data

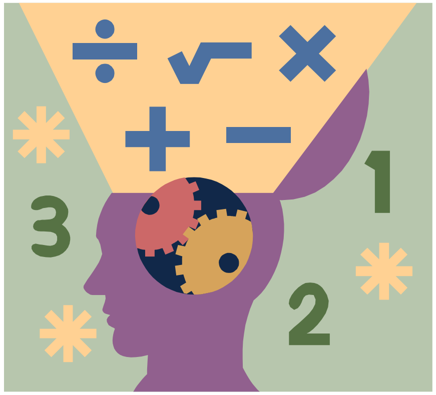
Creating variables in Java



dataType + variableName



Simple data types in Java …..



Price : for example **£4.75** → **A real number**

Total sold: for example **187** → **An integer**

WORDS

T or F

In Java the simple types are referred to as

scalar types or

primitive types

The scalar types of Java

Java type	Allows for	Range of values
`byte`	very small integers	-128 to 127
`short`	small integers	-32768 to 32767
`int`	big integers	-2147483648 to 2147483647
`long`	very big integers	-9223372036854775808 to 9223372036854775807
`float`	real numbers	+/- 1.4 * 10^{-45} to 3.4 * 10^{38}
`double`	very big real numbers	+/- 4.9 * 10^{-324} to 1.8 * 10^{308}
`char`	characters	Unicode character set
`boolean`	true or false	not applicable

The scalar types of Java		
Java type	**Allows for**	**Range of values**
`byte`	very small integers	-128 to 127
`short`	small integers	-32768 to 32767
`int`		-2147483648 to 2147483647
`long`		-9223372036854775808 to 9223372036854775807
`float`	l numbers	+/- 1.4 * 10^{-45} to 3.4 * 10^{38}
`double`	very big real numbers	+/- 4.9 * 10^{-324} to 1.8 * 10^{308}
`char`	characters	Unicode character set
`boolean`	true or false	not applicable

Requires the BIGGEST amount of memory

The scalar types of Java		
Java type	**Allows for**	**Range of values**
`byte`	very small integers	-128 to 127
`short`	small integers	-32768 to 32767
`int`	big integers	-2147483648 to 2147483647
`long`	very big integers	-9223372036854775808 to 9223372036854775807
`floa`	s	+/- 1.4 * 10^{-45} to 3.4 * 10^{38}
`doub`	l numbers	+/- 4.9 * 10^{-324} to 1.8 * 10^{308}
`char`	characters	Unicode character set
`boolean`	true or false	not applicable

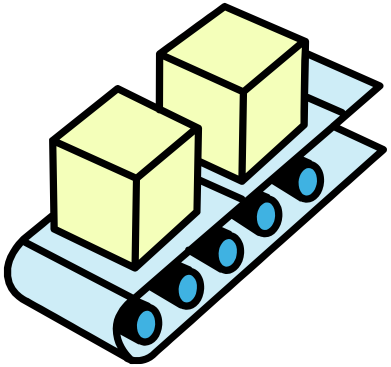
Requires the SMALLEST amount of memory

The scalar types of Java

Java type	Allows for	Range of values
`byte`	very small integers	-128 to 127
`short`	small integers	-32768 to 32767
`int`	big integers	-2147483648 to 2147483647
`long`	very big integers	-9223372036854775808 to 9223372036854775807
`float`	real numbers	+/- 1.4 * 10^{-45} to 3.4 * 10^{38}
`double`	very big real numbers	+/- 4.9 * 10^{-324} to 1.8 * 10^{308}
`char`	characters	Unicode character set
`boolean`	true or false	not applicable

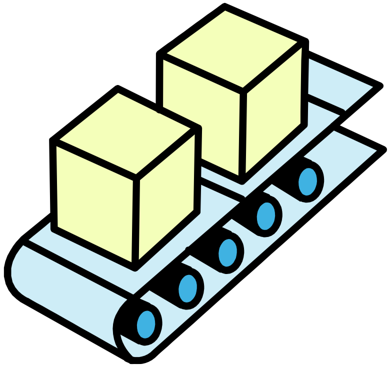
JAVA

```
dataType    variableName ;
```



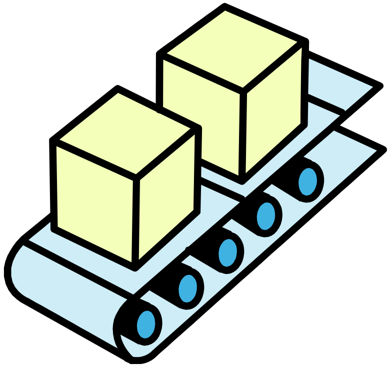
JAVA

```
dataType     age  ;
```



JAVA

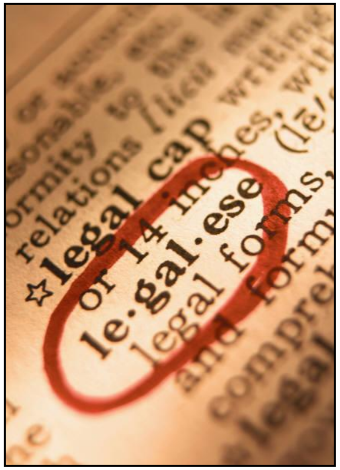
```
int  age  ;
```



You can choose almost any name for your variables as long as:

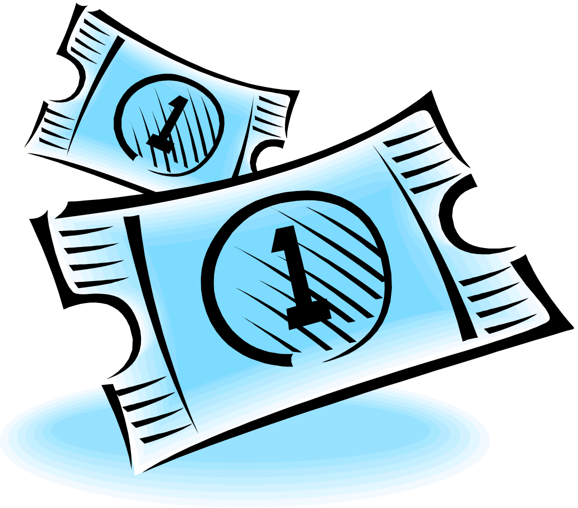
the name **does not have spaces in it**

the name is not already used in Java (such as **class** or **static**)



Activity

“Identify the valid variable names from the following list”



`ticket` ✓

`cinemaTicket` ✓

`void` ✗

`cinema ticket` ✗

`cinema_ticket` ✓

`Ticket` ✓

`ticket` ☑

`cinemaTicket` ☑

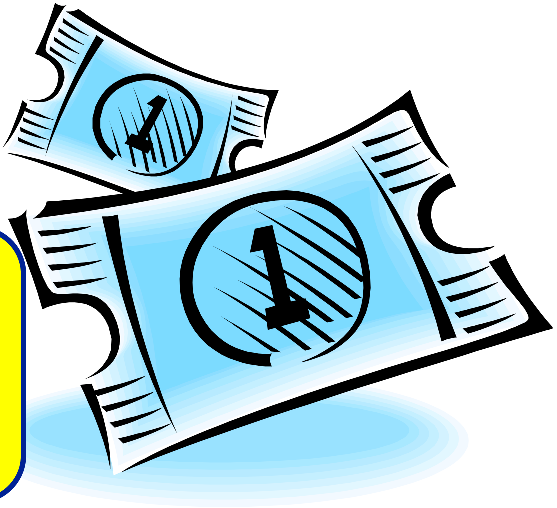
`void` ✗

`cin`

`cin`

`Ticket` ☑

While this name is valid we usually begin variable names with a small letter



So this is preferable.

ticket ☑

cinemaTicket ☑

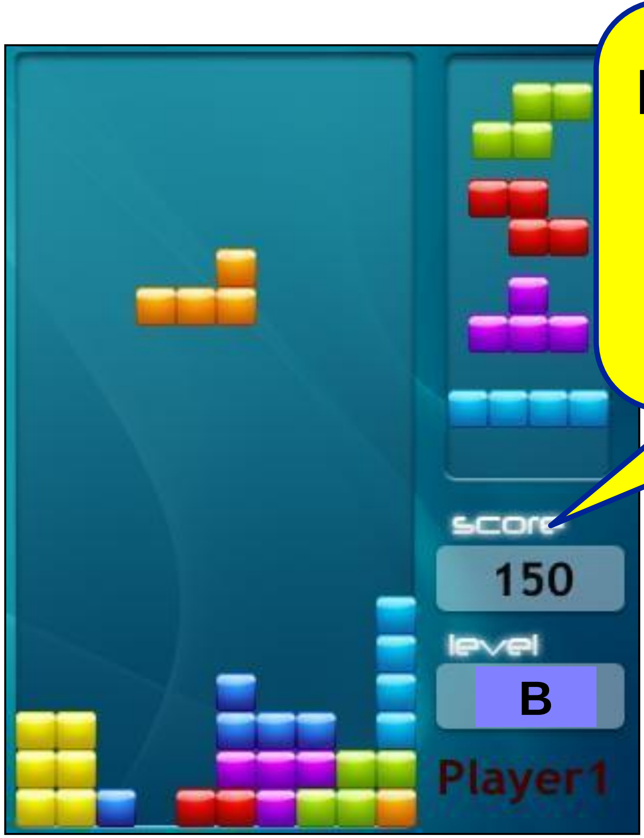
void ✗

cinema ticket ✗

cinema_ticket ☑

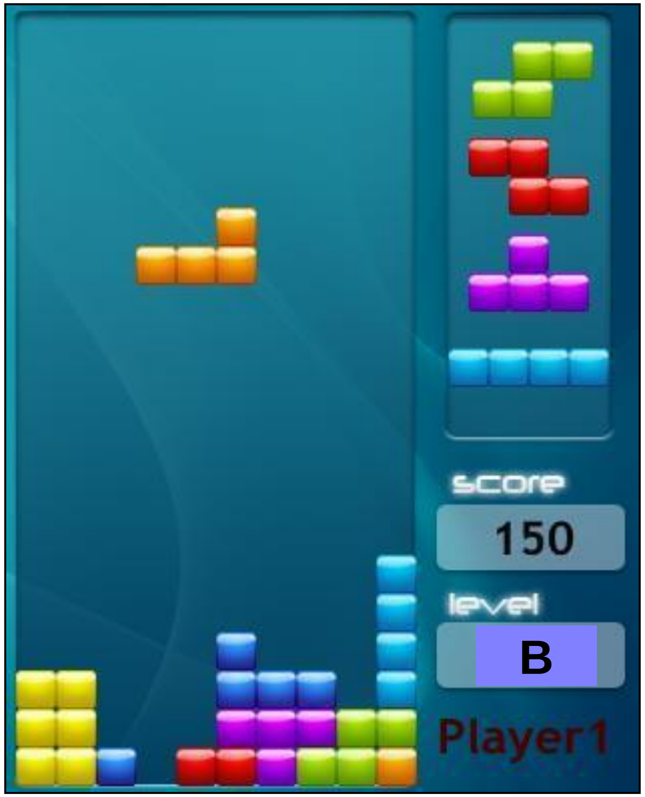
Ticket ☑

Activity



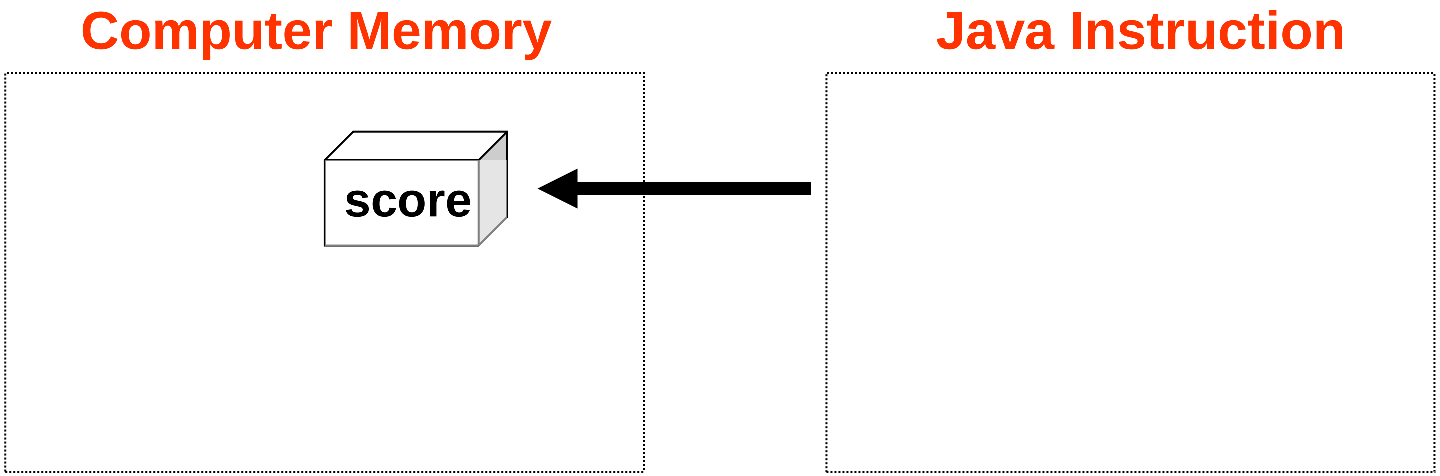
Declare (create) a variable for this item of data

Activity

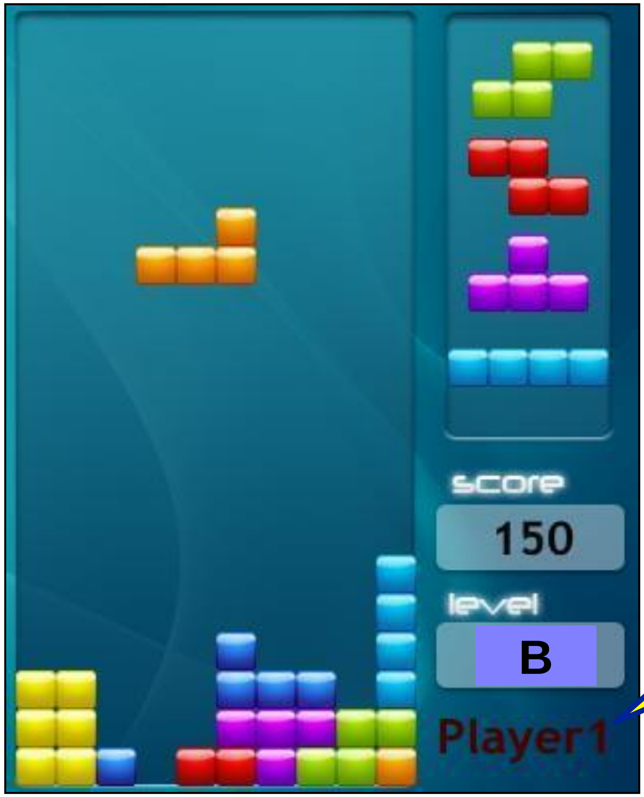


```
int score;
```

The effect of declaring a variable on the computer's memory



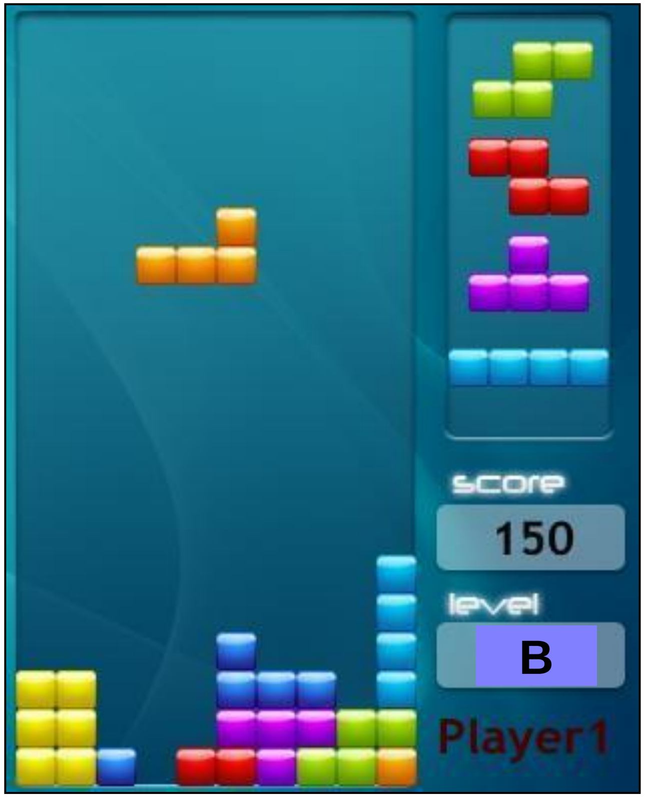
Activity



Declare another variable for this item of data

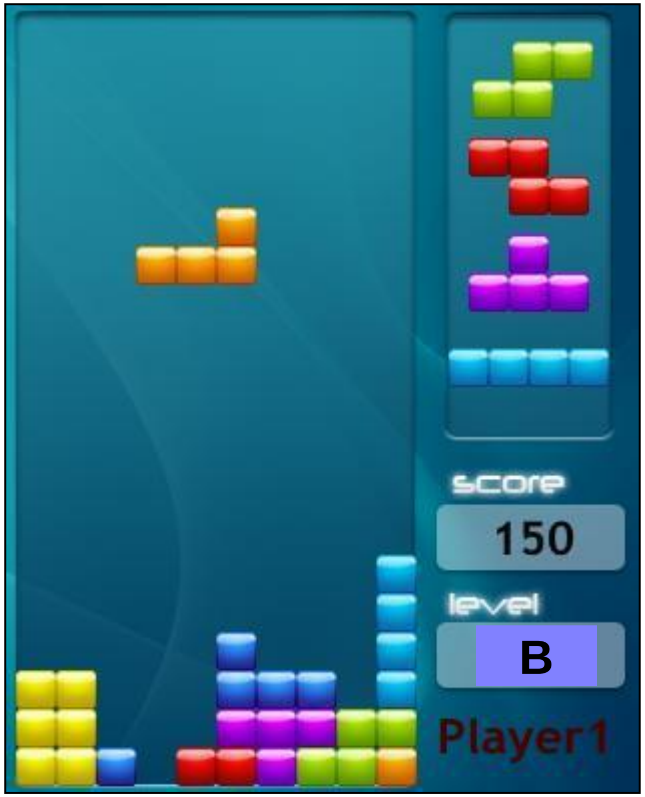
```
int score;
```

Activity



```
int score;
int player;
```

Activity


score

150

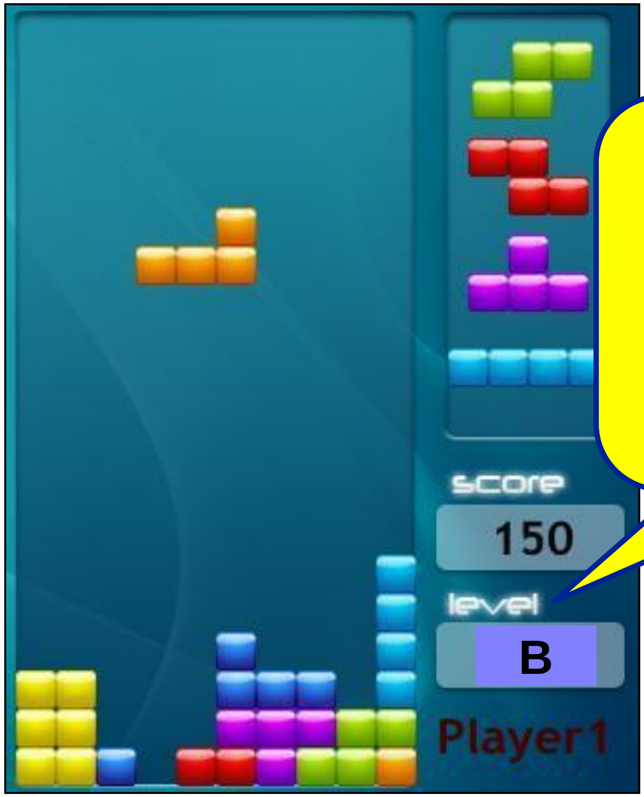
level

B

Player1

```
int score, player;
```

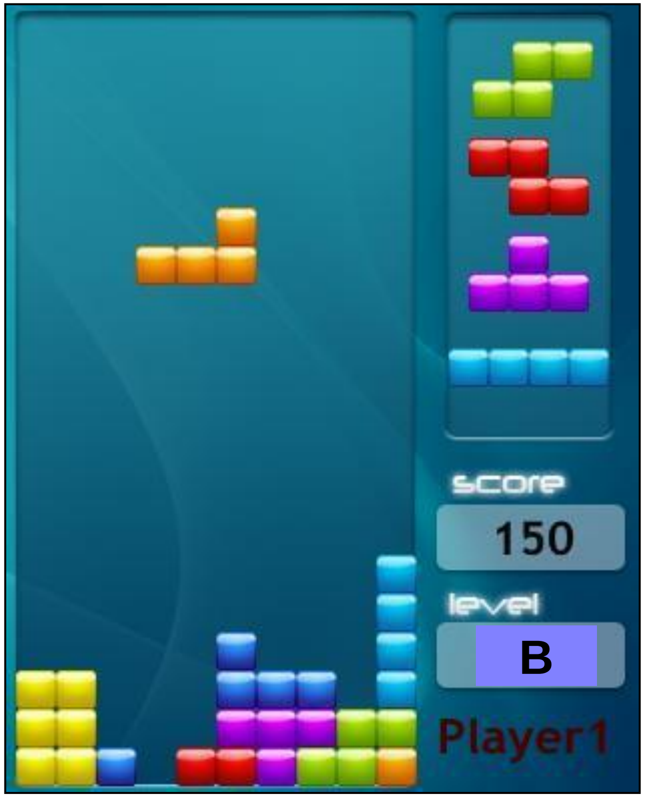
Activity



Declare another variable for this item of data

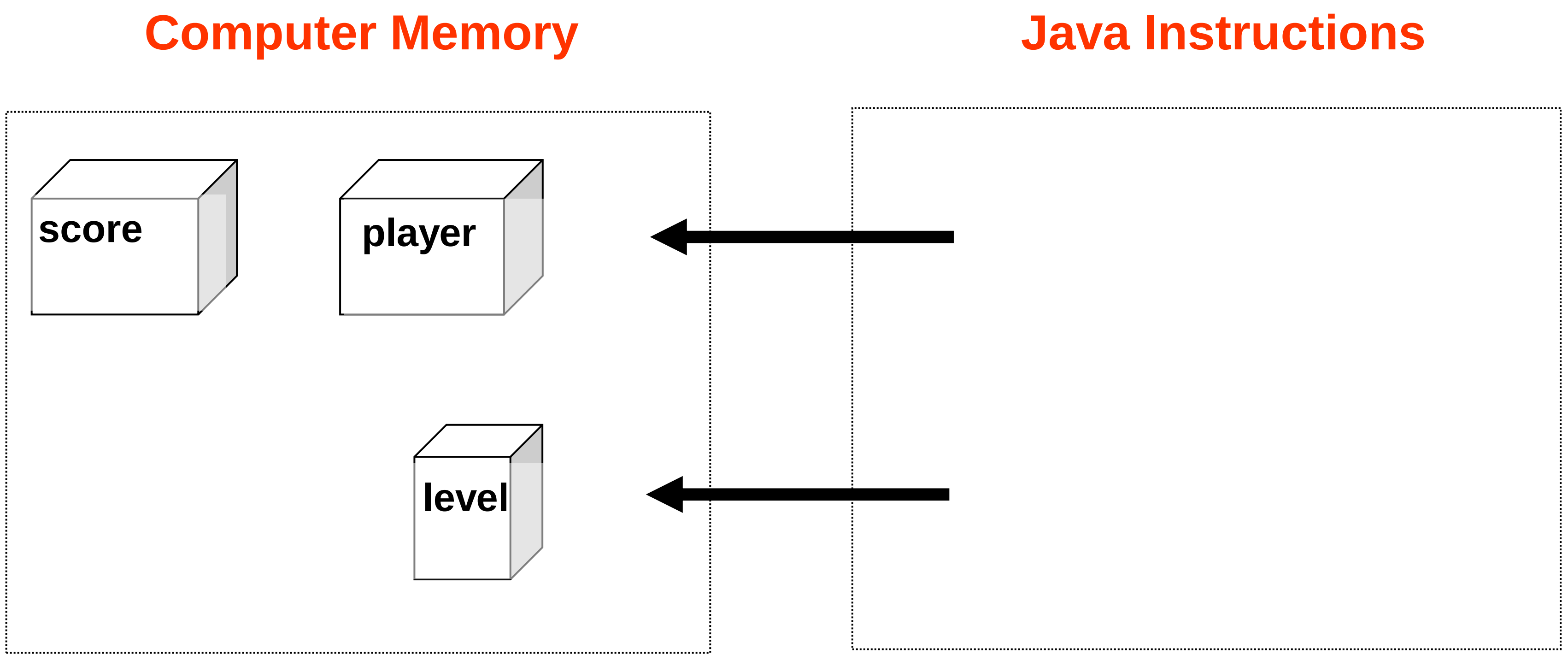
```
int score, player;
```

Activity



```
int score, player;
char level;
```

The effect of declaring many variables in Java



```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

Must declare variables before you can use them!

System.out.println(“***

System.out.print(“Ente

”);

ENTER length

System.out.print(“Enter rectangle height: ”);

ENTER height

System.out.println(“Area of rectangle = ” ??);

Activity

"Declare variables to hold the length and height of the rectangle"

```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

```
double length;
double height;

System.out.println(“*** Rectangle App ***”);
System.out.print(“Enter rectangle length: ”);
          ENTER length
System.out.print(“Enter rectangle height: ”);
          ENTER height
System.out.println(“Area of rectangle = ” ?? );
```

```
*** Rectangle App ***
Enter rectangle length: 7.5
Enter rectangle height: 10
Area of rectangle = 75.0
```

```
double length, height;
```

```
System.out.println("*** Rectangle App ***");
```

```
System.out.print("Enter rectangle length: ");
```

```
ENTER length
```

```
System.out.print("Enter rectangle height: ");
```

```
ENTER height
```

```
System.out.println("Area of rectangle = " ?? );
```

Assignments in Java