

Monday January 14  
Lecture 3

## - Lab 0 Part 2

```
if ( registered ) {
```

go to your registered lab session

```
}  
else {
```

go to Tuesday 5pm session

```
}
```

Preparation  
guide  
soon

- Quiz I Week of Jan 21

- Lab Test I Week of Jan 28

# Data Types

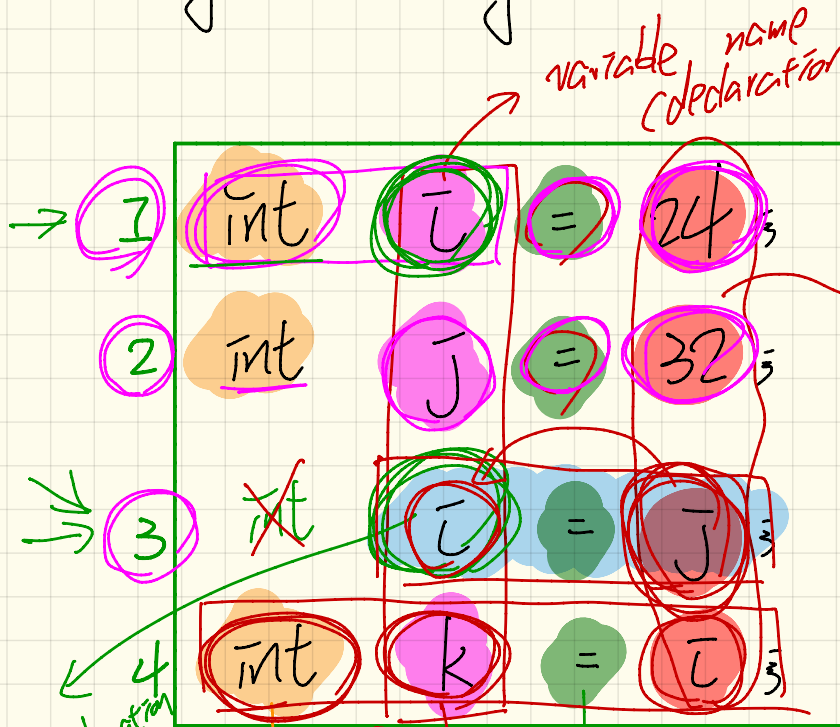
int i = ? ;  
double d = ? ;  
char C = ? ;  
String S = ? ;  
boolean b = ? ;  
↓  
true, false

✓  
23  
i  
"23" X

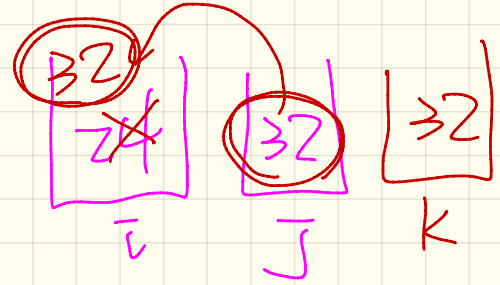
✓  
'c'  
C  
"23" X  
"Hello" X

# Assignment : Change of value

- type
- target
- source



RHS of assignment  
source of assign.



no re-declaration is necessary.

data type

LHS of assign. / target of assign.

assignment

X not compile source

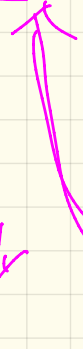
String

target  
name I

= (1+2) \* (23%5);

start  
string  
values only

3  
3  
9



# Constant : Initialization vs. Re-assignment

ConstantCannotBeReassigned.java

```
public class ConstantCannotBeReassigned {  
    public static void main(String[] args) {  
        /* A constant can only be initialized once. */  
        final double pi = 3.14;  
        /* Reassignment of a constant is illegal. */  
        pi = 6.28;  
    }  
}
```

pi can be initialized,  
but cannot re-assigned.

6.28  
~~3.14~~  
pi

# Variable : Initialization vs. Re-assignment

~~3.9~~  
~~5.4~~  
radius 9.6

VariableCanBeReassigned.java

```
public class VariableCanBeReassigned {  
    public static void main(String[] args) {  
        /* A variable can be initialized. */  
        double radius = 5.4;  
        System.out.println("Radius is: " + radius);  
  
        /* A variable may be re-assigned for as many times as necessary */  
        radius = 3.9;  
        System.out.println("Radius is: " + radius);  
        System.out.println("Radius is: " + radius);  
  
        radius = 9.6;  
        System.out.println("Radius is: " + radius);  
    }  
}
```

we can re-assign this var.

5.4

re-assign the variable to a new value.

3.9  
9.6

Expressions (1)

"Hello 5" ← "Hello" + (3+2) → addition force the evaluation first

concat. ←

Type correct?

→ (1 + 2) * (23 % 5)	✓ 9
→ "Hello " + "world"	✓ "Hello world"
→ "Hello" * "world"	✗
→ "Hello " + 3 + 2	✓ "Hello 32"
→ "Hello" * 3	✗
→ "46" % "4"	✗



# EXPRESSIONS (2)

→ ~~["LaLa" + "land" \* (46 % 4)]~~ ✗

→ ("LaLa" + "land" + (46 % 4)) ✓

"LaLa land 2"

2  
↓  
"2"

# Combining Constants and Variables

3.14  
PI

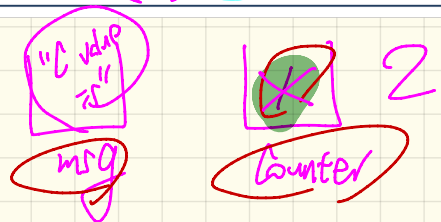
e.g., Print statements involving literals or named constants only:

```
final double PI = 3.14; /* a named double constant */  
System.out.println("Pi is " + PI);  
System.out.println("Pi is " + PI);
```

e.g., Print statements involving variables:

```
String msg = "Counter value is "; /* a string variable */  
int counter = 1; /* an integer variable */  
System.out.println(msg + counter);  
System.out.println(msg + counter);  
counter = 2; /* re-assignment changes variable's stored value */  
System.out.println(msg + counter);
```

re-assignment to msg and counter



# Console Application: With User Inputs vs. Without

With User Input

```
import java.util.Scanner;
public class ComputeAreaWithConsoleInput {
    public static void main(String[] args) {
        /* Create a Scanner object */
        Scanner input = new Scanner(System.in);
        /* Prompt the user to enter a radius */
        System.out.print("Enter a number for radius: ");
        double radius = input.nextDouble();
        /* Compute area */
        final double PI = 3.14169; /* named constant for  $\pi$  */
        double area = PI * radius * radius; /*  $area = \pi r^2$  */
        /* Display result */
        System.out.println(
            "Area for circle of radius " + radius + " is " + area);
    }
}
```

```
public class ComputeArea {
    public static void main(String[] args) {
        double radius; /* Declare radius */
        double area; /* Declare area */
        /* Assign radius */
        radius = 20; /* assign value to radius */
        /* Compute area */
        area = radius * radius * 3.14159;
        /* Display results */
        System.out.print("The area of circle with radius ");
        System.out.println(radius + " is " + area);
    }
}
```

Without User Input

# Common Mistake: Declaring the Same Variable More Than Once

```
int counter = 1;  
int counter = 2;
```

X

re-declaration is not allowed.

Fix 1: Only keep the 1st Declaration

```
int counter = 1;  
counter = 2;
```

[\*] 2  
counter

Fix 2: Declare New Variables

```
int counter1 = 1;  
int counter2 = 2;
```

[1] [2]  
counter1 counter2

# Common Mistake: Using a Variable Before Declaring It

int counter = 1;

variable

- 1 System.out.println("Counter value is " + counter);
- 2 ~~int counter = 1;~~
- 3 counter = 2;
- 4 System.out.println("Counter value is " + counter);

## Example: Convert Seconds to Minutes

```
import java.util.Scanner;
public class DisplayTime {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        /* Prompt the user for input */
        System.out.print("Enter an integer for seconds: ");
        int seconds = input.nextInt();
        int minutes = seconds / 60; /* minutes */
        int remainingSeconds = seconds % 60; /* seconds */
        System.out.print(seconds + " seconds is ");
        System.out.print(" minutes and ");
        System.out.println(remainingSeconds + " seconds");
    }
}
```

Test: 500 seconds

**Exercise:** Modify the program so that it will display hours if necessary. e.g. 7945 seconds → 2 hours  
12 minutes  
9 seconds

# Where Can Assignment Source (RHS) Come From?

In `tar = src`, the *assignment source* `src` may come from:

- A literal

```
int i = 23;
```

- A variable

```
int i = 23;  
int j = i;
```

- An expression involving literals and variables

```
int i = 23;  
int j = i * 2;
```

- An input from the user

```
Scanner input = new Scanner(System.in);  
int i = input.nextInt();  
int j = i * 2;
```