An abstract network diagram with white nodes and lines on a dark blue background, representing a complex system or network.

# CS1101

# Programming and Problem Solving

Dr. Gina Bai

Fall 2022

# Logistics

- **ZY-1** and **ZY-2A** on [zyBook > Assignments](#)
  - Due: Saturday, Jan 21, at 11:59pm
- **PA01 - W, A, B, C** on [zyBook > Chap 11](#)
  - Due: Thursday, Jan 26, at 11:59pm

11.3 PA01-W: String Input/Output (5 pts) Lab

11.4 PA01-A: Read Programming Style Guide Document (5 points) Lab

11.5 PA01-B: Basic Input/Output (15 pts) Lab

11.6 PA01-C: Eiffel Tower (15 points) Lab

# Recap – Expression

A simple value or set of operations that produces a value

- **Operator** → indicates the operation to be performed
- **Operand** → value in the expression

Q: Evaluate the following expressions

- $2 + 2.2 + 2 * 2 + 2$  **10.2**
- $2 + (\text{int}) 2.2 + 2 * 2 + 2$  **10**
- $2 + "(\text{int}) 2.2" + 2 * 2 + 2$  **"2(int) 2.242"**

Step 1: Evaluate  $2 * 2$   
→  $2 + "(\text{int}) 2.2" + 4 + 2$

Step 2: Concatenate  $2 + "(\text{int}) 2.2"$   
→  $"2(\text{int}) 2.2" + 4 + 2$

Step 3: Concatenate  $"2(\text{int}) 2.2" + 4$   
→  $"2(\text{int}) 2.24" + 2$

Step 4: Concatenate  $"2(\text{int}) 2.24" + 2$   
→  $"2(\text{int}) 2.242"$


# More Practice on String Concatenation

- "hi " + "there" → "hi there"
- "hello" + 2023 → "hello2023"
- "abc" + 1 + 2 → "abc12" ("abc" + 1 is evaluated first)
- 1 + 2 + "abc" → "3abc" (1 + 2 is evaluated first)
- "abc" + 9 \* 5 → "abc45"
- "1" + 1 → "11"
- 4 - 1 + "abc" → "3abc"

# The code works, but is redundant

( ) are used to wrap the expressions so that the numbers can be added up first

```
public class Receipt {  
    public static void main(String[] args) {  
        // Calculate total owed, assuming 7% tax and 18% tip.  
        System.out.println("Subtotal: " + (38.0 + 40.0 + 30.0));  
        System.out.println("Tax: " + ((38.0 + 40.0 + 30.0) * 0.07));  
        System.out.println("Tip: " + ((38.0 + 40.0 + 30.0) * 0.18));  
        System.out.println("Total: " + ((38.0 + 40.0 + 30.0) +  
                                         ((38.0 + 40.0 + 30.0) * 0.07) +  
                                         ((38.0 + 40.0 + 30.0) * 0.18)));  
    }  
}
```



# Variables

zyBook Chap 2.3, 2.4



# Variable

A variable is a **memory location** with a **name** and a **type** that stores a **value**.

- E.g., a variable `year` of type `int` and a value of `2023`

`year`

`2023`



# Steps for Using a Variable

1. **Declare** variable
  - Specify its **type** and **name**
2. **Initialize** variable
  - Store a **value** into it
3. **Use** variable
  - Use it as part of an expression/argument

# Step 1 – Declare Variable

- The variable declaration sets aside memory for storing a value
- Syntax: **<type> <varName>;** (stores no value yet)

```
int year;  
boolean isHappy;
```

A diagram showing a green rounded rectangle representing memory. Inside, the variable 'year' is associated with a white rectangular box, and the variable 'isHappy' is associated with a light orange rectangular box.

year  
isHappy

- Naming convention:
  - Start with lowercase letters, capitalize the first letter of the attached words

# Step 2 – Assign Value to Variable

- Variable assignment **stores a value** into a variable
  - The value can be a **number** or an **expression**
  - The **first time** a value is assigned to a variable is also known as **initializing** the variable
- Syntax: **<name> = expression;**
  - read as "<name> **gets** expression" OR "<name> is **assigned** expression"
  - The **=** sign is the command for **assignment**.

```
year = 2023;  
isHappy = true;
```

year	2023
isHappy	true

# Combine Declaration & Initialization

- You can declare and initialize a variable in a single statement

**<type> <name> = expression;**

```
// Approach 1
int year;      // Declare
year = 2023;   // Initialize
```

```
// Approach 2 – Usually preferred
int year = 2023; // Declare and initialize
```

# Step 3 – Use Variable

- Once given a value, a variable can be used in expressions:

```
int x = 4;  
System.out.println("The value of x is: " + x); // The value of x is: 4
```

- We can assign a value to a variable more than once:

```
int x = 4;    // x = 4  
x = 4 + 5;    // x = 9
```

- We can reassign the value based on the variable's current value:

```
int x = 4;    // x = 4  
x = x + 4;    // x = 8
```

The **right-hand side** expression is **evaluated first**, and then its result is **assigned** to the variable **on left**.

Q: Find out the values of the integers, a, b, c, and d.

► JAVA

```
1      int a = 2;
2      int b = 3;
3      int c = 4;
4      int d = a + b + c;    // d = 2 + 3 + 4 = 9
5
6      a = d - a - b;        // a = 9 - 2 - 3 = 4
7      b = d - b - c;        // b = 9 - 3 - 4 = 2
8      c = d - a - c;        // c = 9 - 4 - 4 = 1
9
10     System.out.println("a: " + a);
11     System.out.println("b: " + b);
12     System.out.println("c: " + c);
13     System.out.println("d: " + d);
```

# Scanners

zyBook 1.5



# Scanner

What does a Scanner do?

- **Reads input** from various sources (console, files, etc.) and turns the input into data that can be used by your program.

What is the benefit of a Scanner?

- **Interactive** programs!

# Interactive Programs

- User output: **System.out**
  - `print` and `println` methods print text to the console/terminal
- User input: **System.in**
  - Cannot be used directly
  - Use the `Scanner` class to understand the user's input

# Importing class – Java Class Libraries

- To use Scanner, need to **import** the class from the **Java Class Library**
  - Java Class Library: a set of Java classes available for you to use
    - Classes are organized into groups, which are called **packages**
- **Import declaration** – goes at the top of your program file
  - `import <package name>.*;`
  - `import <package name>.<class name>;`

# Importing class – Scanner

- Requires import
  - `import java.util.*;` OR
  - `import java.util.Scanner;`
- Construction of Scanner for console
  - `Scanner <name> = new Scanner(System.in);`
    - Common names: input, console, scnr, ...

# Tokens

A **single** element of input (e.g., **one word, one number**)

- The Scanner object reads in user input as tokens
- **Tokens** are separated by **whitespace**, e.g.,
  - Space
  - Tab
  - Newline character '\n'

# Scanner Methods

Methods that can be run on **Scanner** objects.

Method	Description
<code>nextInt()</code>	Reads and returns user input as an <b>int</b>
<code>nextDouble()</code>	Reads and returns user input as a <b>double</b>
<code>next()</code>	Reads and returns user input as a <b>String</b>
<code>nextLine()</code>	Reads and returns and <b>entire line</b> of user input as a <b>String</b>

- Methods **wait** for the user to type the input and press <Enter>
- Value typed by the user is returned to your program
- You want to **prompt the user for input**

```
import java.util.Scanner;

/**
 * Class gets input from users on the number of credit hours they are currently taking.
 *
 * @author Gina Bai
 */
public class CreditHour {

    public static void main(String[] args) {
        // Construct the scanner, which is called input, for the input from console
        Scanner input = new Scanner(System.in);

        // Prompt user for credit hours
        System.out.println("How many credit hours are you currently taking?");

        // Read in the number of credit hours as an integer, and stored it to int creditHour
        int creditHour = input.nextInt();

        // Print message
        System.out.println("You are currently taking " + creditHour + " credit hours.");
    }
}
```

**Always do**  
**Step 1: prompt for input**  
**Step 2: read in the input**



# Error Handling

- `InputMismatchException`
  - If the next token **does not match** the pattern for **the expected type**, or is out of range for the expected type
- `NoSuchElementException`
  - If the input is exhausted

# Escape Sequences

zyBook Chap 2.8

# Escape Sequence

- A two-character sequence starting with a backslash \ that represents a special character
- Has special meaning to the compiler

Escape Sequences	Description
\t	Insert a <b>tab</b> in the text at this point
\n	Insert a <b>newline</b> in the text at this point
\'	Insert a <b>single quote</b> character in the text at this point
\"	Insert a <b>double quote</b> character in the text at this point
\\	Insert a <b>backslash</b> in the text at this point

# In-class Coding Practice

- Brightspace > Lectures > Milestone 1 > Lecture 5

## Lecture 5 - Variables & Scanners & Escape Sequences ▾

 Print

Class Meeting: 01/20/2023

Topics: String concatenation, Variables (Declare - Initialize - Use), Scanners for console input, Escape sequences

Assignments and Deadlines:

- ZY-1 & ZY-2A on zyBook > Assignments, due Saturday, Jan 21, at 11:59pm
- PA01 - W, A, B, C on zyBook > Chap 11, due Thursday, Jan 26, at 11:59pm

 Download

EscapeSeq ▾  
 JAVA File

```

public class EscapeSeq {

    public static void main(String[] args) {

        System.out.println("=== Using \\" ==");
        System.out.println( "\"Computers are incredibly fast, accurate, and stupid. " +
            "Human beings are incredibly slow, inaccurate, and brilliant. " +
            "Together they are powerful beyond imagination.\"- Albert Einstein");

        System.out.println("=== Using '\\' ==");
        System.out.println("char is surrounded with single quotes ('')");

        System.out.println("=== Using \\n ==");
        System.out.println("Hello\\nWorld");

        System.out.println("=== Using \\t ==");
        System.out.println("Hello\\tWorld");

        System.out.println("=== Using double backslashes \\ to print a single backslash \\ ==");

    }
}

```

# Sample Solution

```

$ javac EscapeSeq.java
$ java EscapeSeq
=== Using \
"Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow,
inaccurate, and brilliant. Together they are powerful beyond imagination.\"- Albert Einstein
=== Using \'
char is surrounded with single quotes (')
=== Using \\n
Hello
World
=== Using \\t
Hello World
=== Using double backslashes \\ to print a single backslash \

```