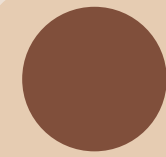


# Topic 1.5: Getting & Using User Input

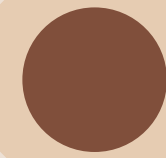
# Learning Goals (Week 1):



Identify data types based on value



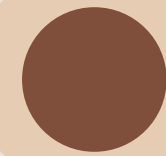
Map variables to the current values



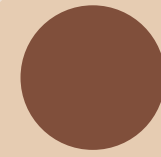
Perform basic operations on variables



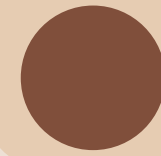
Create and use Java and user-defined methods



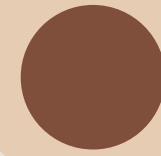
Format Printed Output



**Obtain and process user input from the console**



Use booleans, conditionals, and compound conditionals correctly



Select and implement different types of loops depending on scenario



Use special String and Math operations



Successfully implement and manipulate java arrays

# How do we get user input?

- **Scanner** can be used to get input (keyboard input) from the user during the execution of a program
- Very useful if you need to interact/prompt the user for some information

## Java “Libraries”

- Just like a real library, it would be crazy pants to bring every book possible home every time you wanted to read
- There is a base set of behaviours we get when we install Java but that isn't everything...
- You only “borrow” what you actually need from the extras
- We now need user input capabilities so we need to “borrow” (**import**) this book from the java library
- At the top of your program, outside your methods/classes:

```
import java.util.Scanner;
```

# Now to “open” the Scanner “book”

- The **import** statement above is like bringing the book home from the library. We can now read it whenever we want, but we have to **open** the book and sit down and read to get anything from it.
- In the method(s) you want to use the Scanner “book” and read from the user you need to declare a Scanner variable:

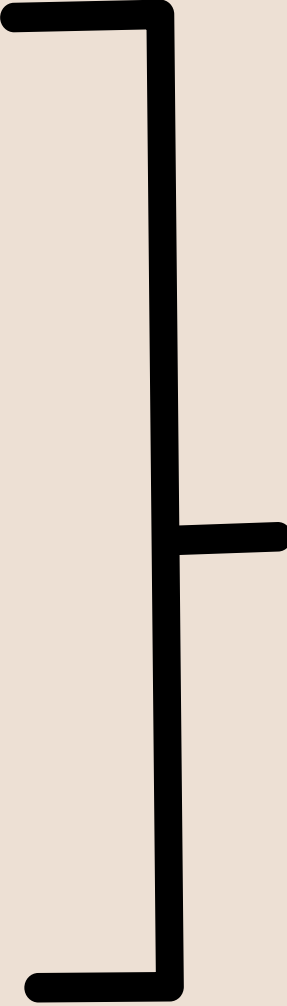
```
Scanner sc = new Scanner(System.in); // sc is now the variable we can use
```



System.in means read input from the console

# Scanner methods we can use

keyboardInput.nextInt()	-> <b>int</b>
keyboardInput.nextLong()	-> <b>long</b>
keyboardInput.nextFloat()	-> <b>float</b>
keyboardInput.nextDouble()	-> <b>double</b>
keyboardInput.next()	-> <b>String</b>
keyboardInput.nextLine()	-> <b>String</b>



Return only the next **token**: sequence of non-blank characters  
*means the new line character (or extra spaces etc. are still on the line and need to be processed)*

# Scanner methods we can use

keyboardInput.nextInt()      -> **int**  
keyboardInput.nextLong()      -> **long**  
keyboardInput.nextFloat()      -> **float**  
keyboardInput.nextDouble()      -> **double**  
keyboardInput.next()      -> **String**

keyboardInput.nextLine()      -> **String** } Returns the entire line, blanks and all

**These methods** automatically convert the keyboard input to the specified type. You will get an error if the next token entered by the user is not of the expected format.

# Type Conversions

We can use type conversions to **explicitly** convert from one data type to another

## String to Primitive type conversion:

Integer.parseInt(String) -> int  
Long.parseLong(String) -> long  
Double.parseDouble(String) -> double  
Float.parseFloat(String) -> float  
Boolean.parseBoolean(String) -> boolean

Replace String in the above methods by any String you want to convert.

Once again, the String must be convertible to the corresponding type, otherwise you'll get an error.

## Primitive to String conversion

Integer.toString(int)  
Long.toString(long)  
Double.toString(double)  
Float.toString(float)  
Boolean.toString(boolean)

Called automatically by Java when you concatenate (+) a primitive and a String

**Don't forget we also learned by type casting as a way of forcing data type conversion for *some* types**

# Pause & Practice

- Ask the user to type in a number between 0 - 99. Store the input into an **int value** variable and then print the value.
- Ask the user to type a decimal number between 0 - 1. Store it in a **double value** variable and then print the value entered. Following this, multiply the stored value by 10. Finally, convert this new value to an integer and print it out. This process involves capturing user input and performing basic type conversion in Java.
- Ask the user to type in a number between 0 - 99. Store the input into a **String input** variable. **Convert** the input variable value to an integer and store it in an **int convertInput** variable. Print this variable.