

# Inputs in Java



UNIVERSITY OF  
CALGARY

---

CPSC 319 - Data Structures

Benyamin Bashari

# Reading Input in Java

- ▷ Reading Input From Console
- ▷ Reading Command Line Arguments
- ▷ Reading Input From File

# Reading Input From Console

- ▷ Scanner Class
- ▷ You need to import it from `java.util.Scanner`
- ▷ Initialize it with `System.in`
  - `Scanner sc = new Scanner(System.in)`
- ▷ Use methods to read
  - `sc.next()` -> Reads a Token
  - `sc.next[Byte/Short/Int/Long/Double/Float/Boolean]()`
  - `sc.nextLine()`
  - `sc.setDelimiter(String pattern)`

# Reading Command Line Arguments

- ▷ Run the java file with the arguments you want
  - `$java Main (arg 1) (arg 2) .... (arg n)`
- ▷ The method main has a parameter, `String[] args`
  - `public static void main(String[] args)`
  - Each String in the array represent one of the arguments, seperated by space

# Reading Input From File

## Method 1 : Scanner

- ▷ Again we can use Scanner class to read from a file.
- ▷ But now we also need File Class to read a file.
- ▷ `File f = new File(String path)`
  - Import from `java.io.File`
- ▷ Most of the classes in Java that works with input and output with files, use Exceptions to show different unwanted events regards reading and writing from/to a file.

# Exception

- ▷ Unexpected events in the program
- ▷ If you do not handle the exceptions your program terminates, this is not what we want
- ▷ Two Types of Exception
  - Checked: Must be handled in the code, otherwise compilation error, like `IOException`
  - Unchecked: You can ignore it, like `ArithmeticException`
- ▷ There are two ways to handle exceptions in Java
  - Try-Catch-Finally
  - Throw the exceptions from method
  - See `ExceptionExample.java`

# Reading Input From File

## Method 1 : Scanner

- ▷ See `ReadFileExample.java`
- ▷ Other methods of Scanner
  - `hasNext()`
  - `hasNext[Byte/Short/Int/Long/Double/Float/Boolean]()`

# Reading Input From File

## Method 2 : BufferedReader

- ▷ See `ReadFileExample.java`
- ▷ Methods
  - `close()`
  - `lines()`
  - `read()` -> single char
  - `readLine()`



# String Class

## ▷ Constructors

- `String()`
- `String(char[] values)`
- Lot More!!!

## ▷ Methods

- `charAt(int index)`
- `compareTo(String t)`
- `concat(String t)`
- `equals(String t)`
- `indexOf(int ch, [int fromIndex])`
- `toCharArray()`
- Lot More!!!

# String Class

## ▷ Constructors

- `String()`
- `String(char[] values)`
- Lot More!!!

## ▷ Methods

- `charAt(int index)`
- `compareTo(String t)`
- `concat(String t)`
- `equals(String t)`
- `indexOf(int ch, [int fromIndex])`
- `toCharArray()`
- Lot More!!!

# Exercise

- ▷ Part of the second assignment
- ▷ Read a string from arguments (fileName)
- ▷ Read contents of the file
- ▷ The file consists of words, each word in a separate line
- ▷ Sort each word alphabetically (Not Bubble Sort)