Standard Input
&
Decisions

COMP2026

PROBLEM SOLVING USING OBJECT ORIENTED PROGRAMMING

Overview

- Standard Input
- If statements
- Switch statements

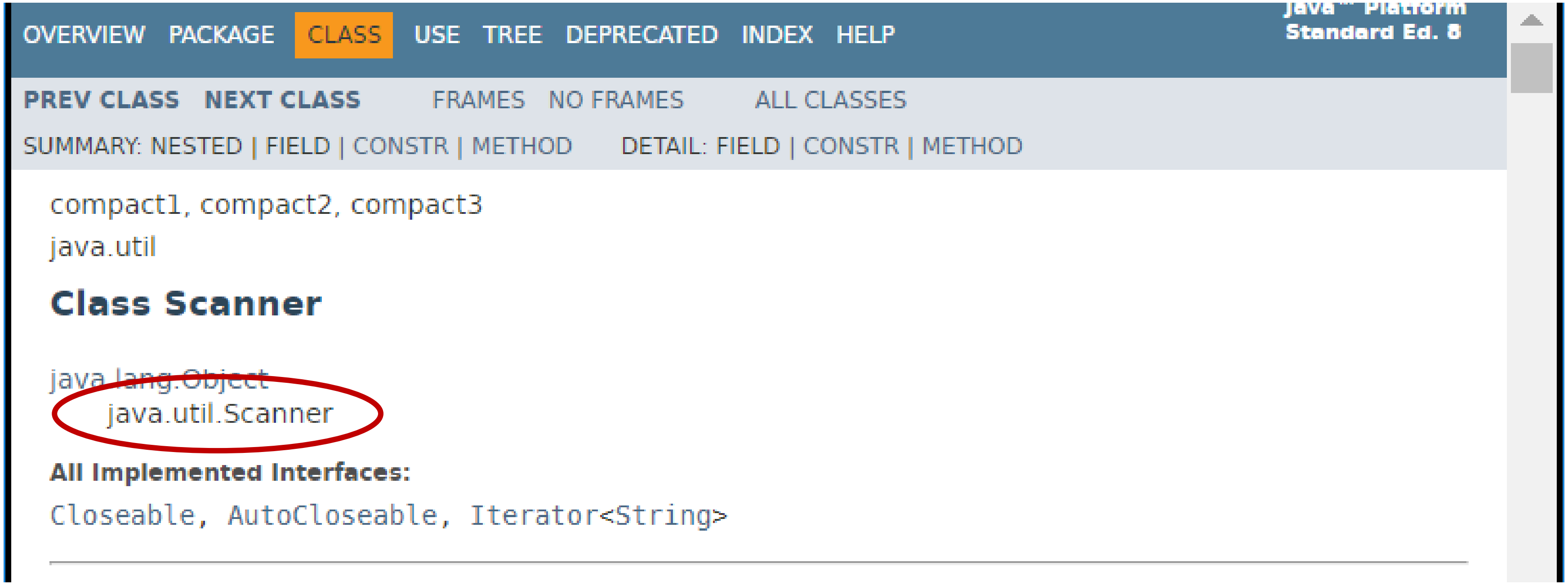
Reading input

- A program has to read keyboard input by a class called Scanner inside the Java API (libraries)
- Classes in Java API are grouped in packages
- A package is a collection of classes with a related purpose
- Scanner class is inside a package called java.util
- http://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html

Reading input

- To use the Scanner class, we have to
  1. Import the class from its package at the top of the program

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



compact1, compact2, compact3

java.util

Class Scanner

java.lang.Object

java.util.Scanner

All Implemented Interfaces:

Closeable, AutoCloseable, Iterator<String>

Reading input

- To use the Scanner class, we have to

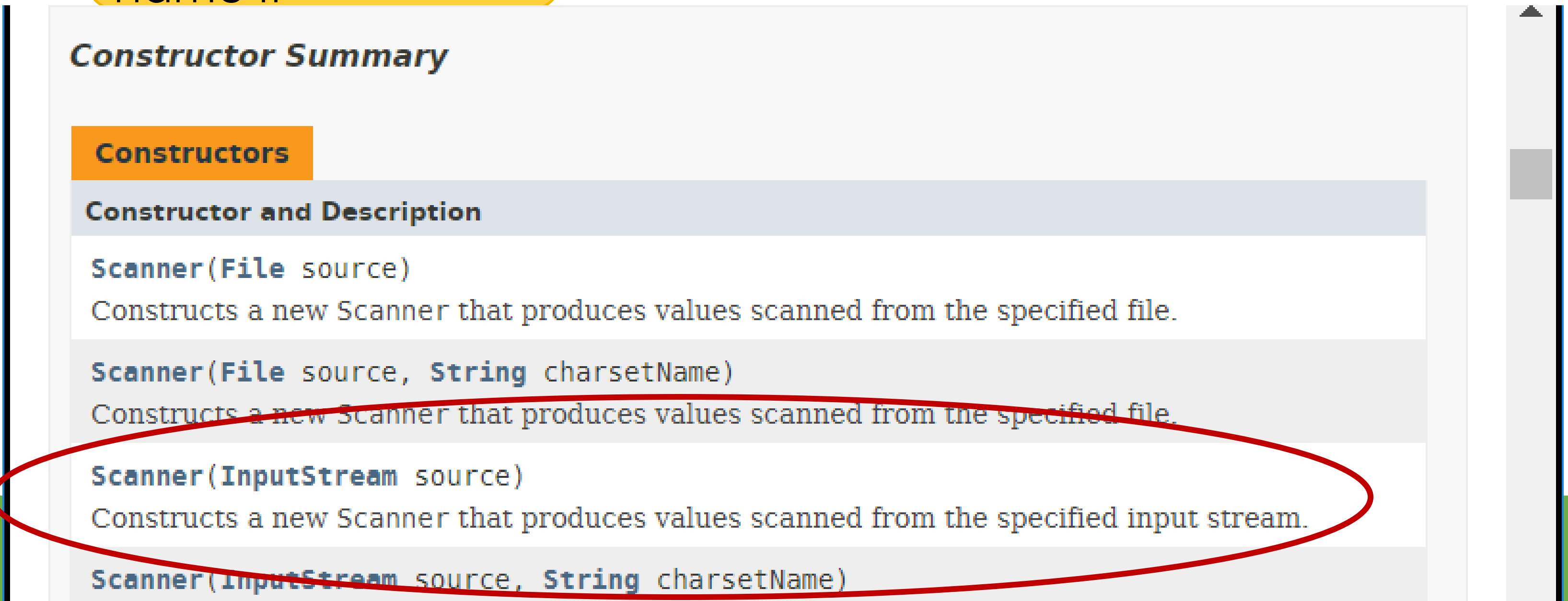
2. Obtain a Scanner object by

```
Scanner in = new Scanner(System.in);
```

InputStream

object name
Similar to variable name, you could name it

to create a new object



Constructor Summary

Constructors

Constructor and Description
`Scanner(File source)` Constructs a new Scanner that produces values scanned from the specified file.
`Scanner(File source, String charsetName)` Constructs a new Scanner that produces values scanned from the specified file.
`Scanner(InputStream source)` Constructs a new Scanner that produces values scanned from the specified input stream.
`Scanner(InputStream source, String charsetName)`

Reading input

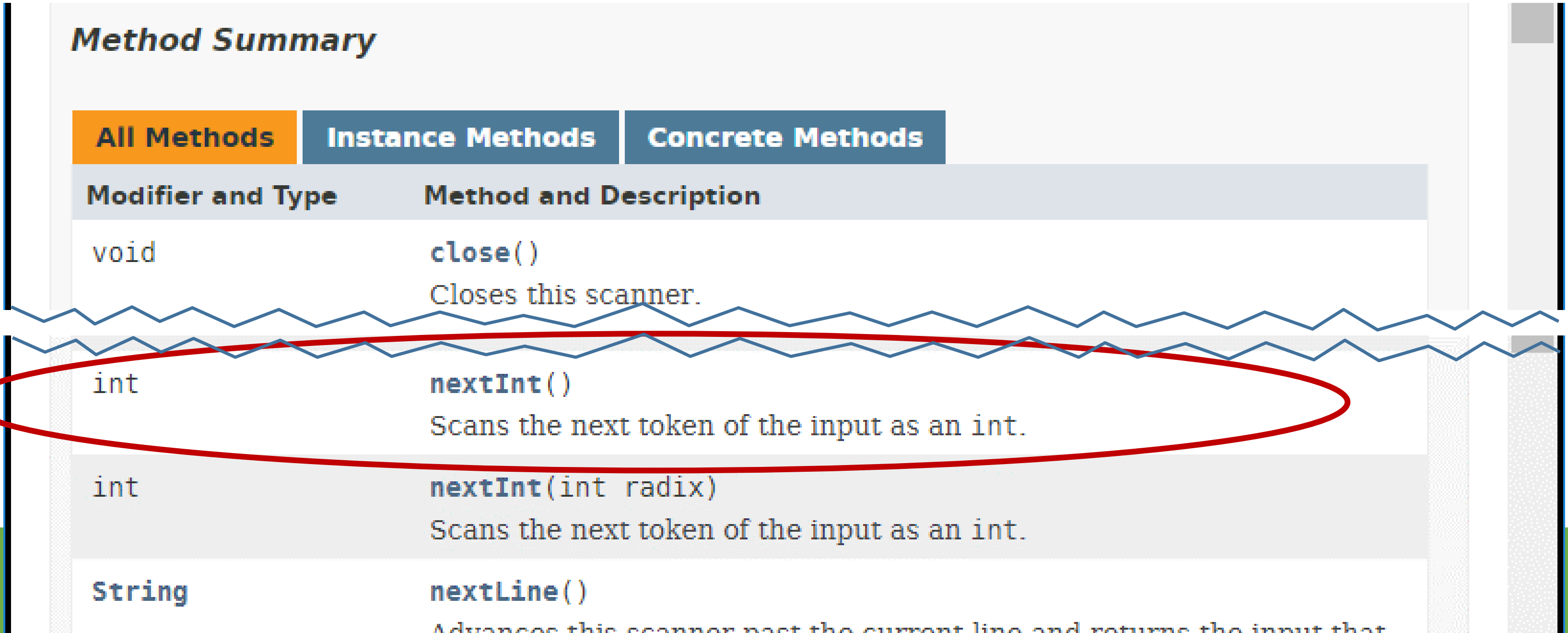
- Once we have a scanner, we use its nextInt() method to read an integer value

a period (.)

```
int num1 = in.nextInt();
```

object name

method to read an integer



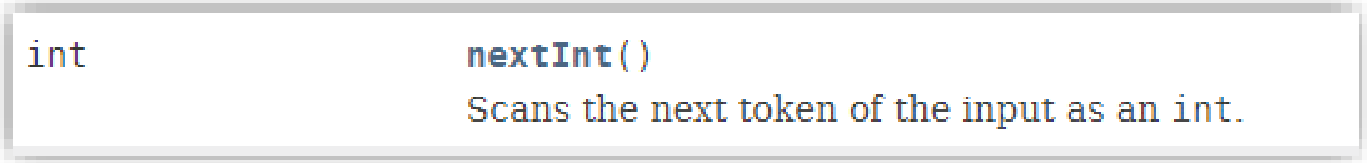
Method Summary

All Methods | Instance Methods | Concrete Methods

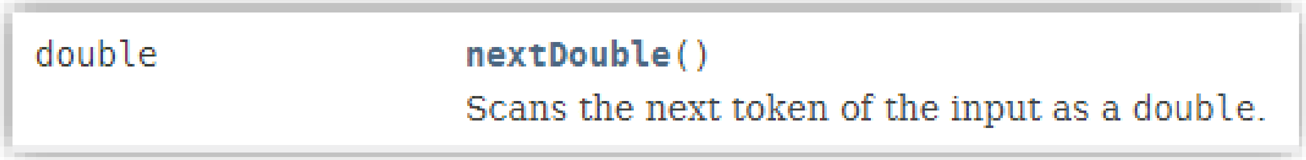
Modifier and Type	Method and Description
`void`	`close()` Closes this scanner.
`int`	`nextInt()` Scans the next token of the input as an int.
`int`	`nextInt(int radix)` Scans the next token of the input as an int.
`String`	`nextLine()` Advances this scanner past the current line and returns the input that

Reading input

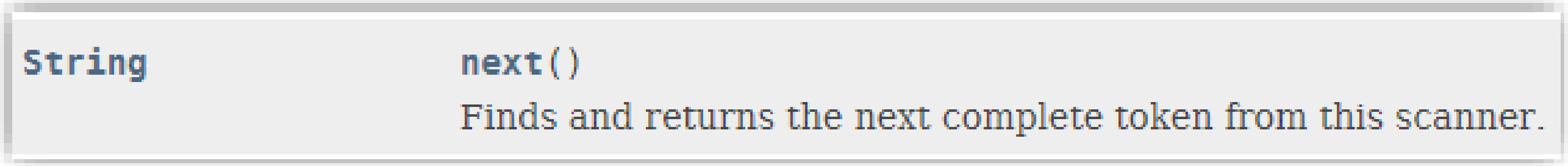
- Common input methods:



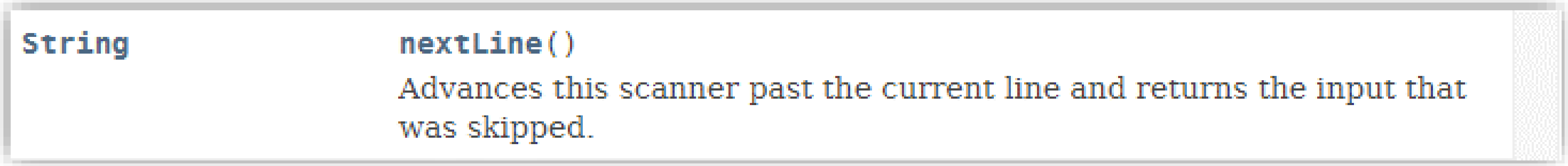
int	**nextInt()** Scans the next token of the input as an int.



double	**nextDouble()** Scans the next token of the input as a double.



String	**next()** Finds and returns the next complete token from this scanner.



String	**nextLine()** Advances this scanner past the current line and returns the input that was skipped.

```
int n = in.nextInt();            // read input as integer
double d = in.nextDouble();      // read input as double
String str = in.next();          // read input as String
String line = in.nextLine();     // read whole line as String
```

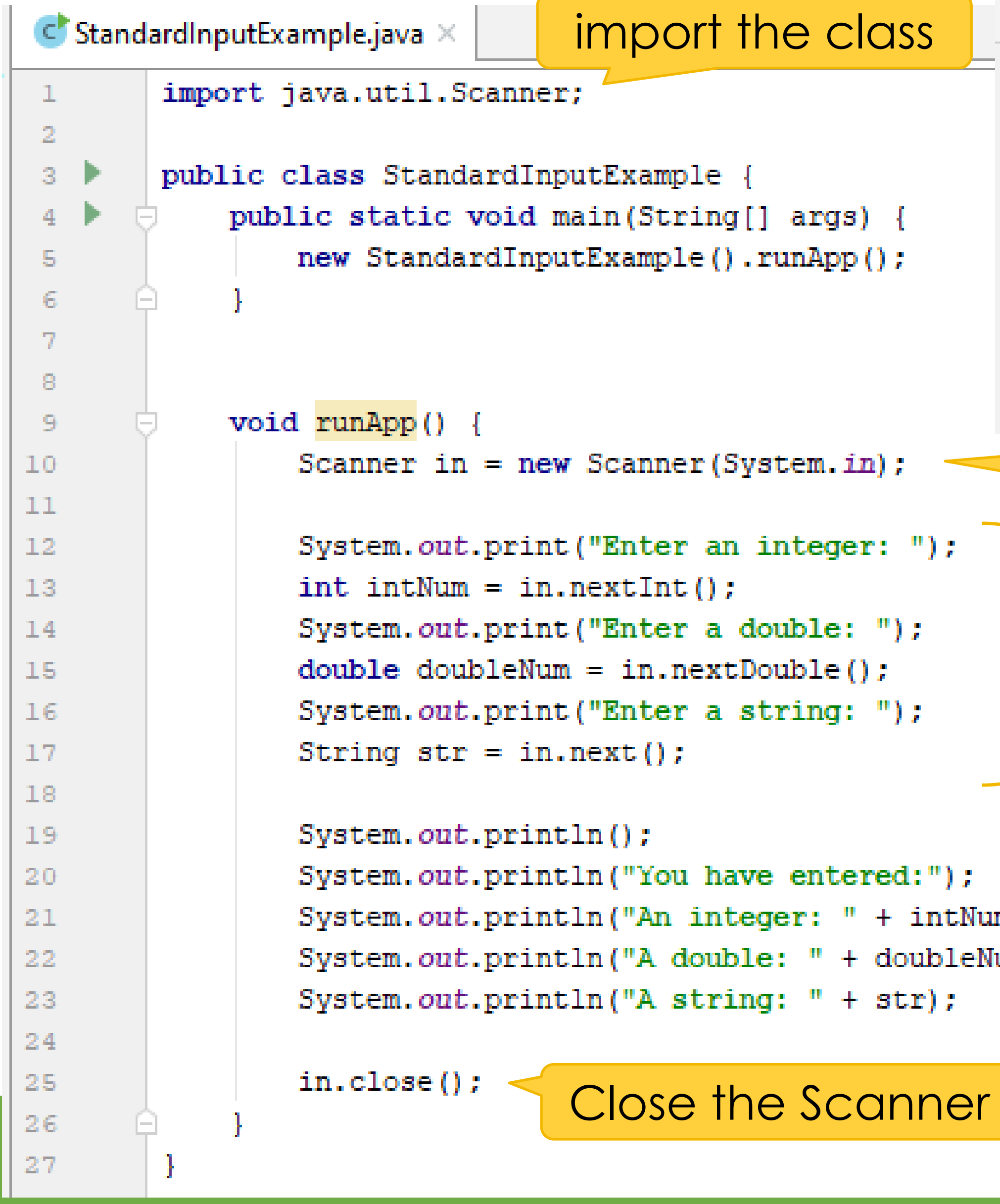
Reading input

- When a program asks for user input, it should first print a message that tells the user which input is expected
- Such a message is called **prompt**
- Example:

```
System.out.print(“Please input a number: ”);
```

use the print method

Standard Input Example



```java
import java.util.Scanner;

public class StandardInputExample {
    public static void main(String[] args) {
        new StandardInputExample().runApp();
    }


    void runApp() {
        Scanner in = new Scanner(System.in);

        System.out.print("Enter an integer: ");
        int intNum = in.nextInt();
        System.out.print("Enter a double: ");
        double doubleNum = in.nextDouble();
        System.out.print("Enter a string: ");
        String str = in.next();

        System.out.println();
        System.out.println("You have entered:");
        System.out.println("An integer: " + intNum);
        System.out.println("A double: " + doubleNum);
        System.out.println("A string: " + str);

        in.close();
    }
}
```

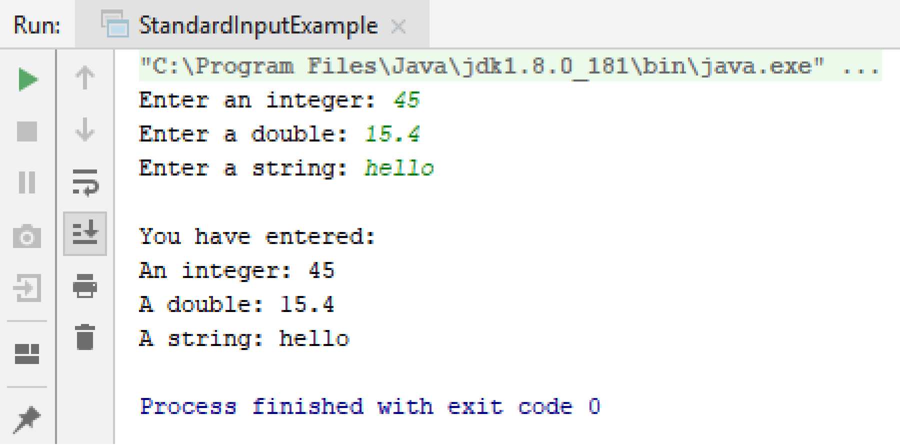
import the class

Create a Scanner object

Display prompts and read inputs

Print result

Close the Scanner

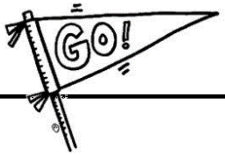


```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Enter an integer: 45
Enter a double: 15.4
Enter a string: hello

You have entered:
An integer: 45
A double: 15.4
A string: hello

Process finished with exit code 0
```

If statements

Recap
Writing conditions & doing selections

Relational / Equality Operators	Meaning
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
!=	not equals
==	equals



```
if (condition)
{
    the operations that you want
    to do if condition is TRUE
}
```

```
if (condition)
{
    the operations that you want
    to do if condition is TRUE
}else
{
    the operations that you want
    to do if condition is FALSE
}
```

```
if (condition 1)
{
        operations to do if condition 1 is TRUE
}else if (condition 2)
{
        operations to do if condition 2 is TRUE
}else if (condition 3)
{
        operations to do if condition 3 is TRUE
}else if (condition n)
{
        operations to do if condition 4 is TRUE
}else
{
        operations to do if ALL the above
        conditions are FALSE
}
```

Recap on if

```
…
//read input number
Scanner in = new Scanner(System.in);
System.out.print("Enter 1, 2, or 3: ");
int input = in.nextInt();

if (input == 1) {
    System.out.println("You entered 1.");
} else if (input == 2) {
    System.out.println("You entered 2.");
} else if (input == 3) {
    System.out.println("You entered 3.");
} else {
    System.out.println("That’s not 1, 2, or 3!");
}
...
```

Switch statements

Recap on switch

```
…
//read input integer
Scanner in = new Scanner(System.in);
System.out.print("Enter 1, 2, or 3: ");
int input = in.nextInt();

switch(input)
{
 case 1:
    System.out.println("You entered 1.");
    break;
 case 2:
    System.out.println("You entered 2.");
    break;
 case 3:
    System.out.println("You entered 3.");
    break;
 default:
    System.out.println("That’s not 1, 2, or 3!");
}
...
```

Recap on switch

```
…
//read input character
Scanner in = new Scanner(System.in);
System.out.print("Enter a letter: ");
char input = in.next().charAt(0);

switch(input)
{
 case 'a':
 case 'e':
 case 'i':
 case 'o':
 case 'u':
    System.out.println("Vowel!");
    break;
 default:
    System.out.println("Consonant!");
}
...
```

0	1	2	3	4
h	e	l	l	o

Recap on switch

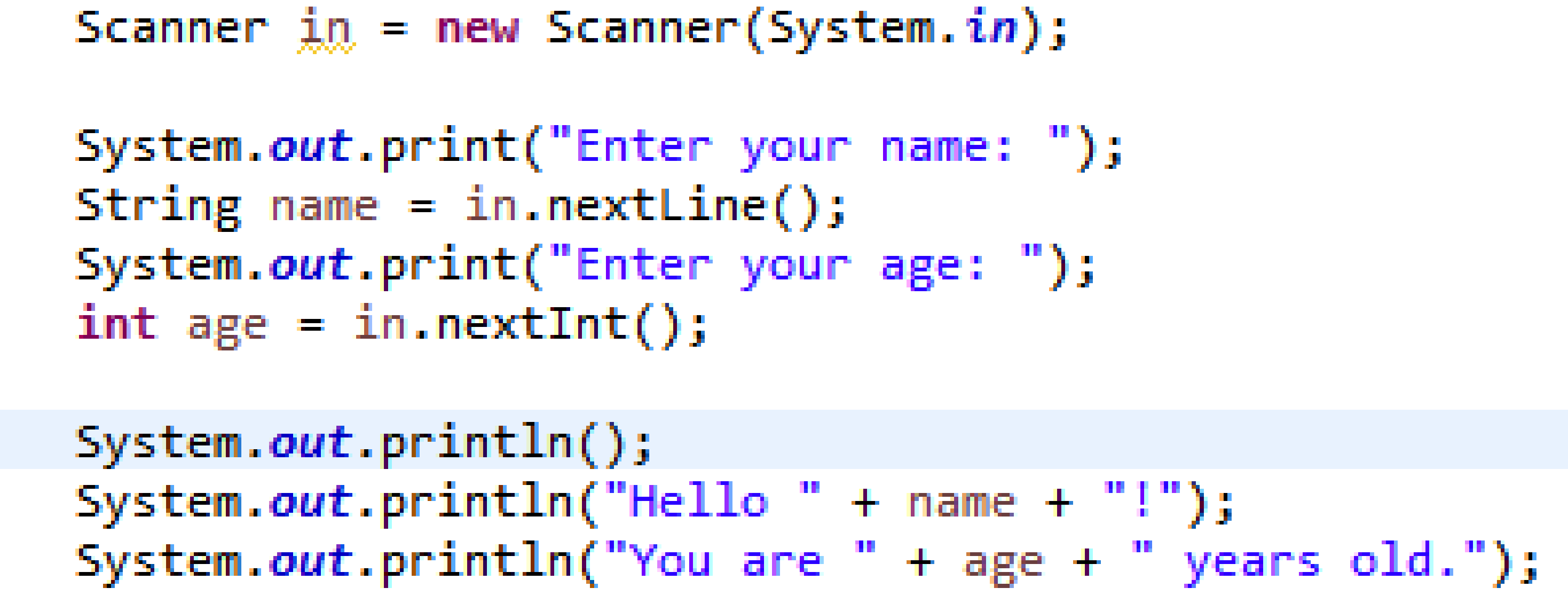
```
…
//read input String
Scanner in = new Scanner(System.in);
System.out.print("Enter a month: ");
String input = in.next();

switch(input)
{
  case "January":  case "March":     case "May":
  case "July":     case "August":  case "October":
  case "December":
    System.out.println("31 days");
    break;
  case "April":          case "June":
  case "Septemer":       case "November":
    System.out.println("30 days");
    break;
  case "February":
    System.out.println("28 or 29 days");
    break;
  default:
    System.out.println("Error!");
}
. . .
```

Part A
Discovery Exercises

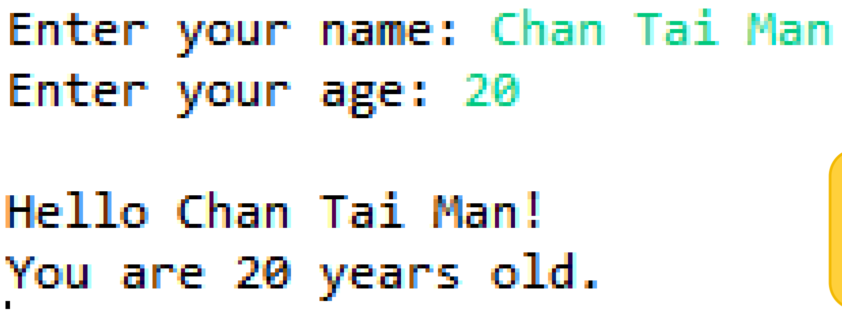
Type your answers in **XXXXXXXX_lab02.docx**

Hints for Task 1

More about reading input



```
Scanner in = new Scanner(System.in);

System.out.print("Enter your name: ");
String name = in.nextLine();
System.out.print("Enter your age: ");
int age = in.nextInt();

System.out.println();
System.out.println("Hello " + name + "!");
System.out.println("You are " + age + " years old.");
```

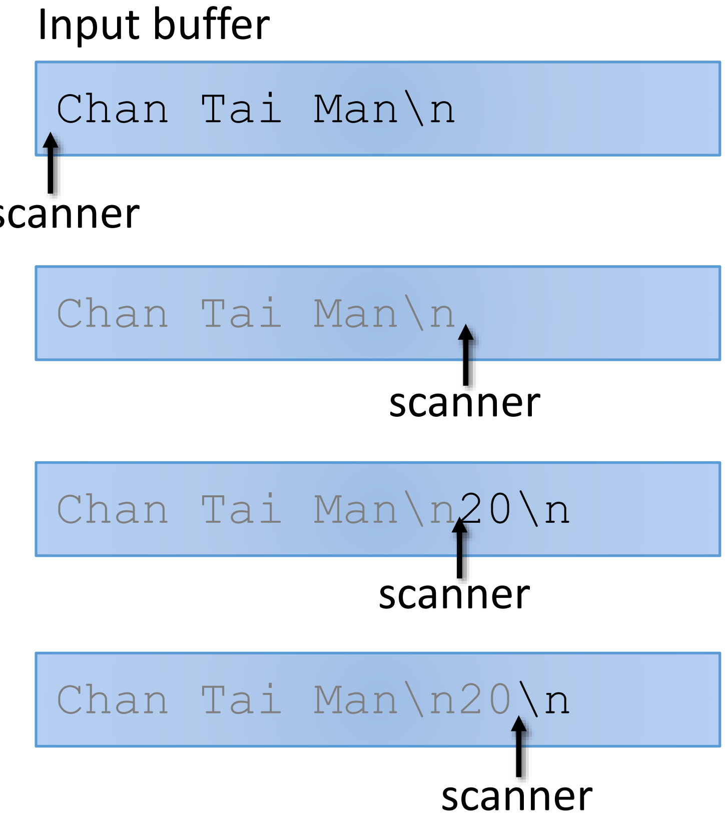


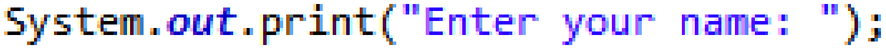
```
Enter your name: Chan Tai Man
Enter your age: 20

Hello Chan Tai Man!
You are 20 years old.
```

Let's see how the input was read by the computer…

More about reading input

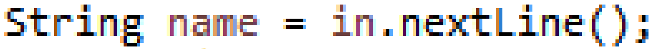
- Understanding how input was read





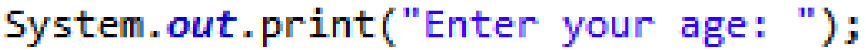
```
System.out.print("Enter your name: ");
```

nextLine() reads input up to the \n

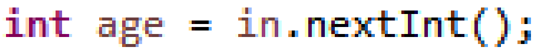


```
String name = in.nextLine();
```

name stores “Chan Tai Man”
Scanner advances to next line (after \n)



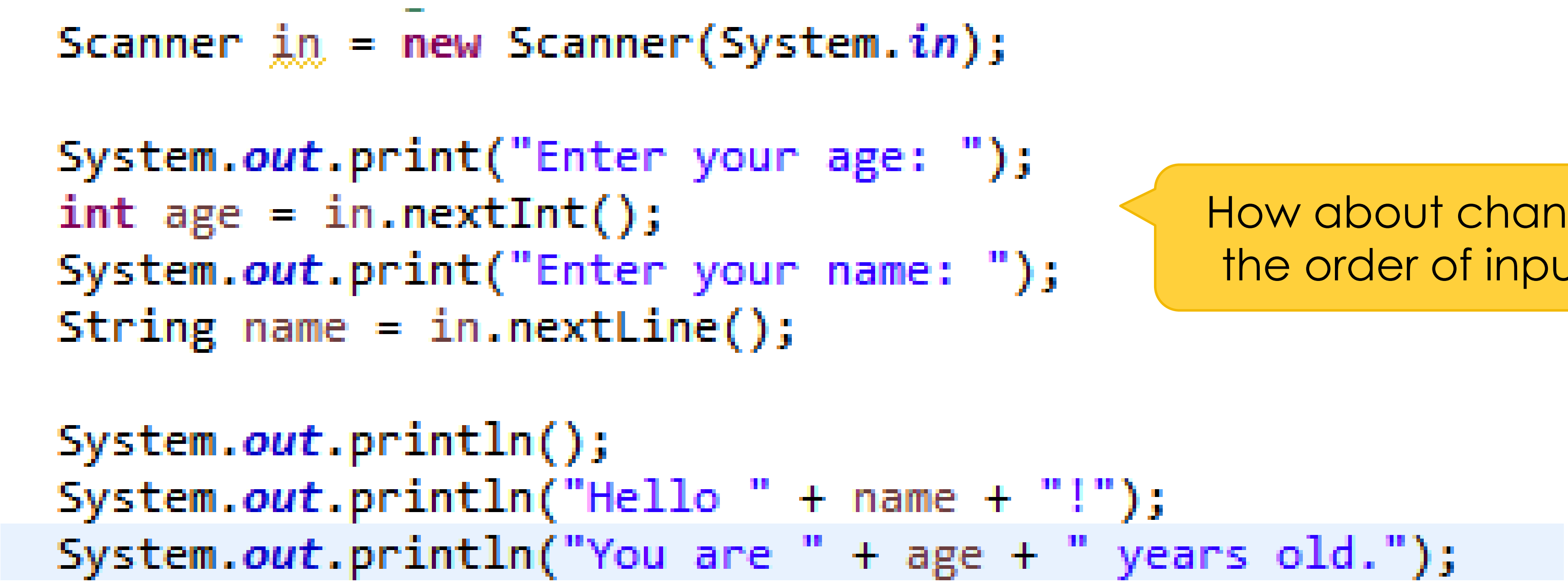
```
System.out.print("Enter your age: ");
```



```
int age = in.nextInt();
```

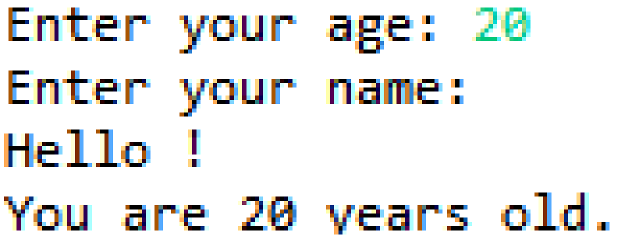
nexitInt() takes 20
\n is left in the input buffer

More about reading input



```
Scanner in = new Scanner(System.in);

System.out.print("Enter your age: ");
int age = in.nextInt();
System.out.print("Enter your name: ");
String name = in.nextLine();

System.out.println();
System.out.println("Hello " + name + "!");
System.out.println("You are " + age + " years old.");
```

How about changing the order of inputs ?

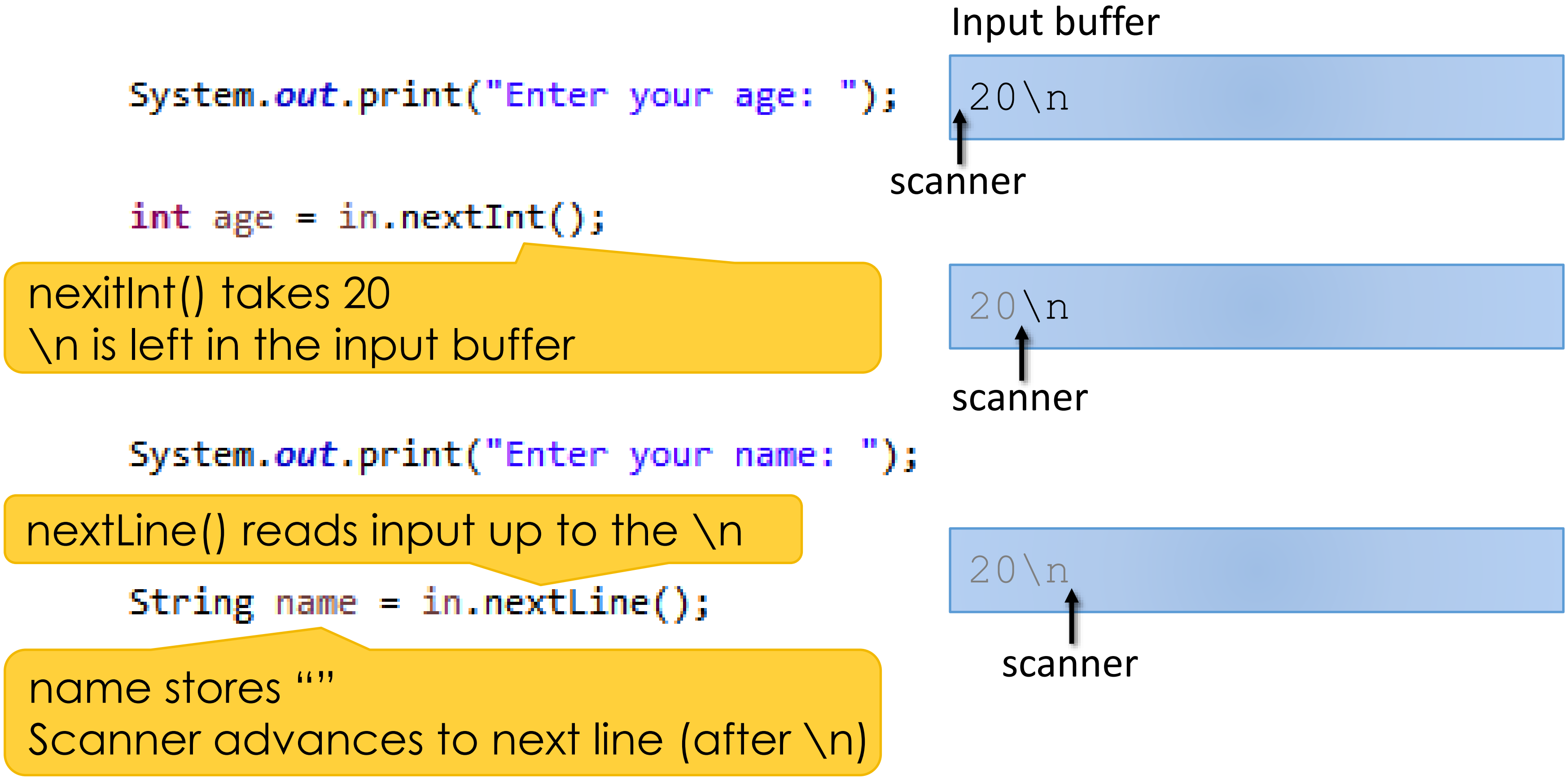


```
Enter your age: 20
Enter your name:
Hello !
You are 20 years old.
```

What’s wrong?
After typing 20, the program prints the output directly!!

More about reading input

- Understanding how input was read



```
System.out.print("Enter your age: ");

int age = in.nextInt();
```

nexitInt() takes 20
\n is left in the input buffer

```
System.out.print("Enter your name: ");
```

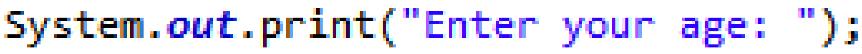
nextLine() reads input up to the \n

```
String name = in.nextLine();
```

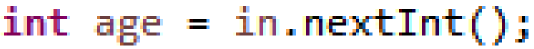
name stores “”
Scanner advances to next line (after \n)

Solution

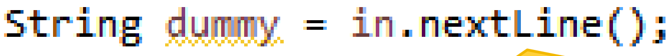
- Add a nextLine() method to solve the problem



```
System.out.print("Enter your age: ");
```

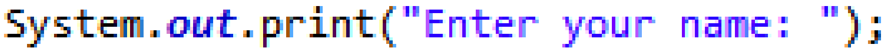


```
int age = in.nextInt();
```

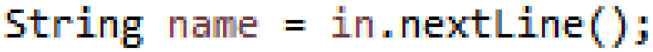


```
String dummy = in.nextLine();
```

dummy store “”
Scanner advances to next line (after \n)

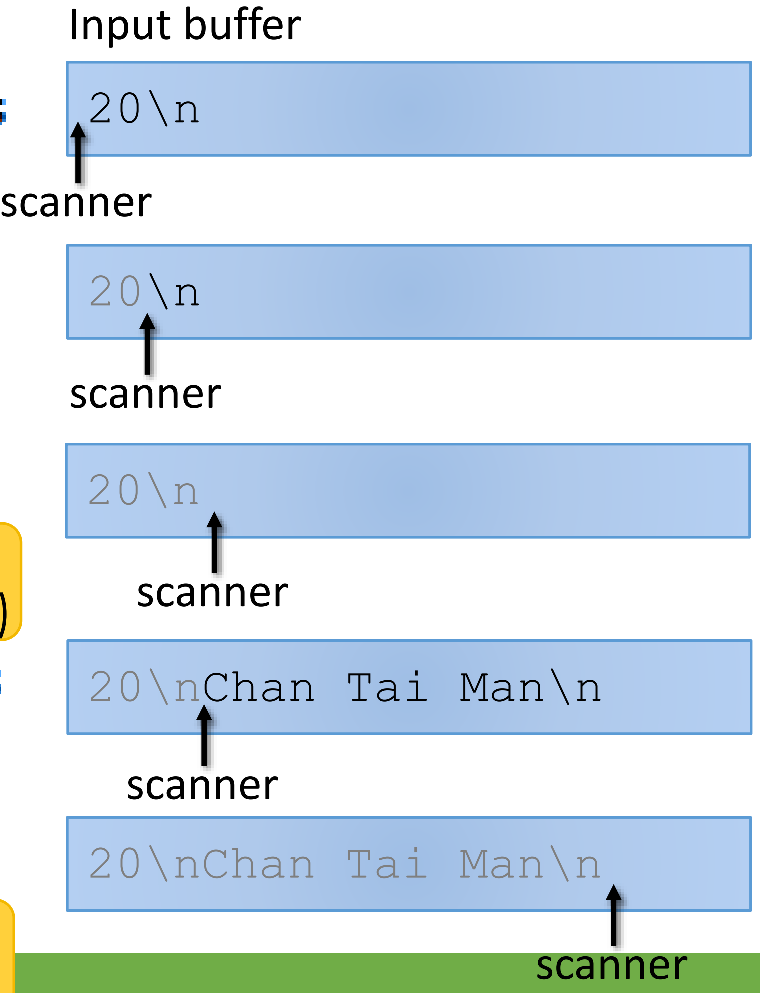


```
System.out.print("Enter your name: ");
```

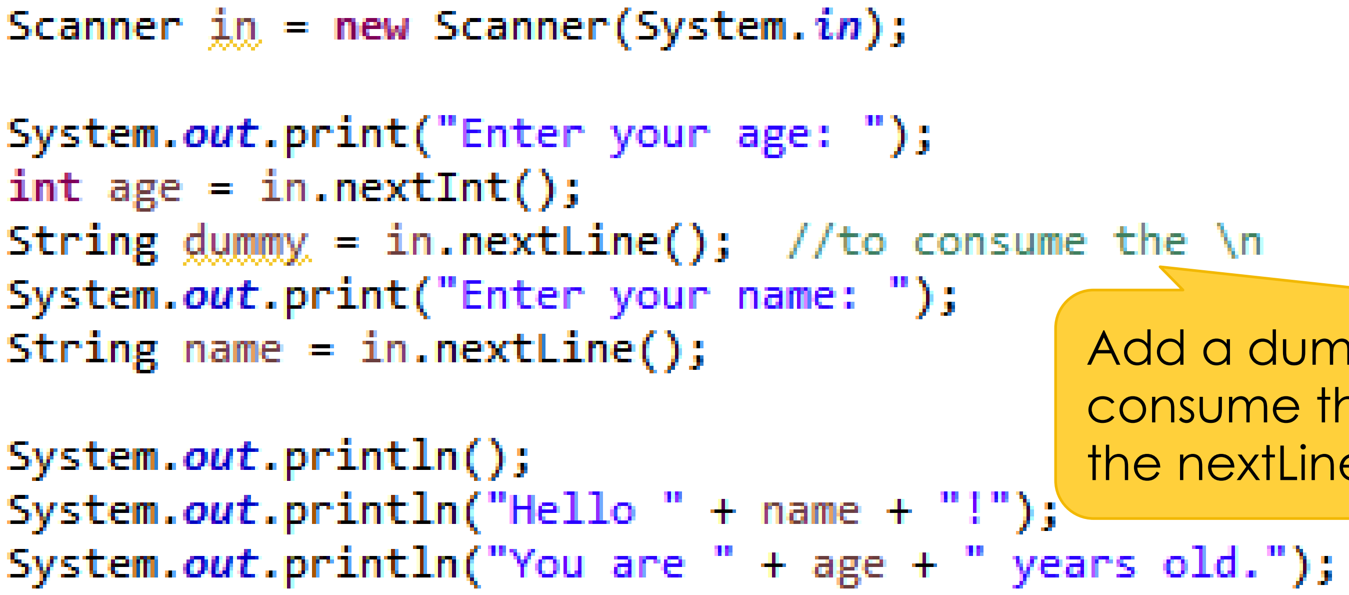


```
String name = in.nextLine();
```

name stores “Chan Tai Man”
Scanner advances to next line (after \n)

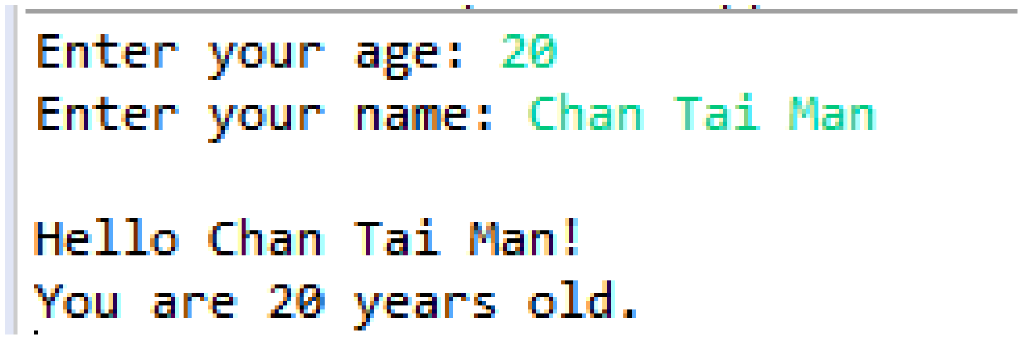


Solution



```
Scanner in = new Scanner(System.in);

System.out.print("Enter your age: ");
int age = in.nextInt();
String dummy = in.nextLine();  //to consume the \n
System.out.print("Enter your name: ");
String name = in.nextLine();

System.out.println();
System.out.println("Hello " + name + "!");
System.out.println("You are " + age + " years old.");
```

Add a dummy nextLine() to consume the \n before using the nextLine to read input

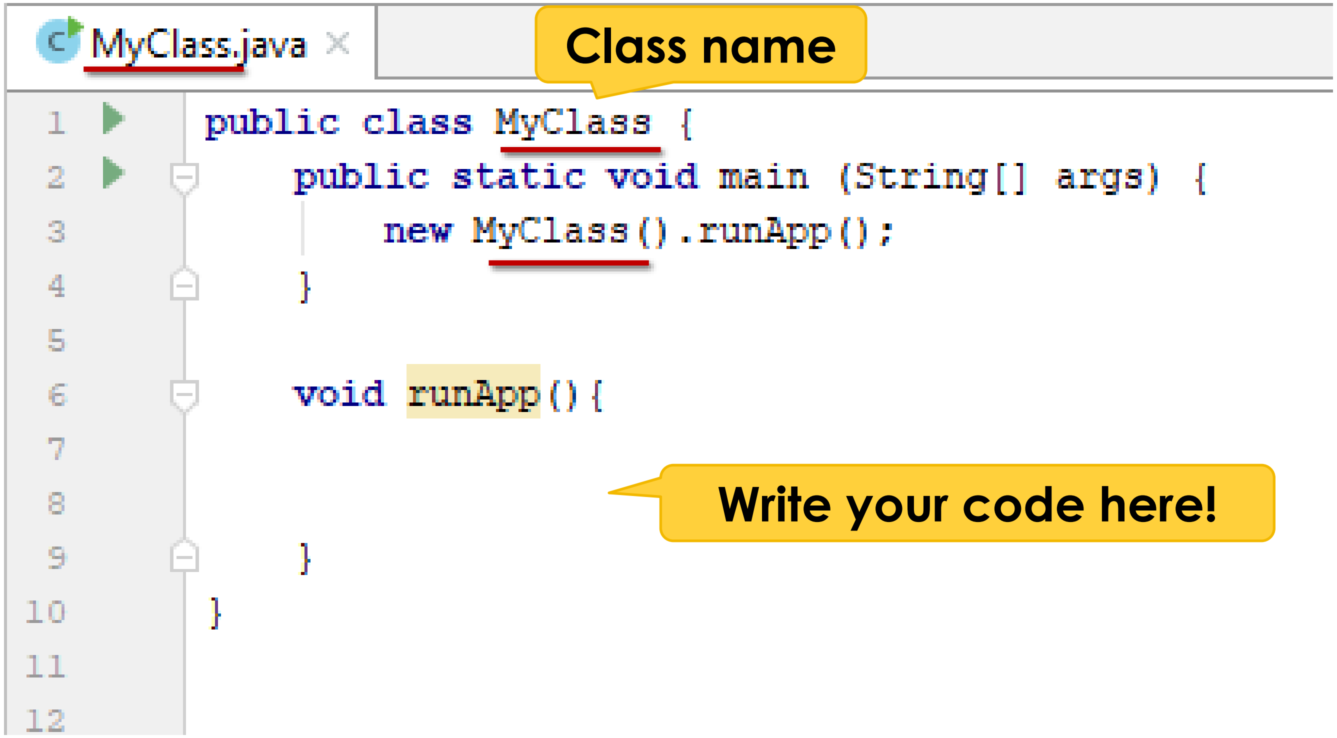


```
Enter your age: 20
Enter your name: Chan Tai Man

Hello Chan Tai Man!
You are 20 years old.
```

Part B
Programming Exercises

How to start?

- Suppose you are told to write a Java program called MyClass



```
public class MyClass {
    public static void main (String[] args) {
        new MyClass().runApp();
    }

    void runApp(){


    }
}
```

Class name

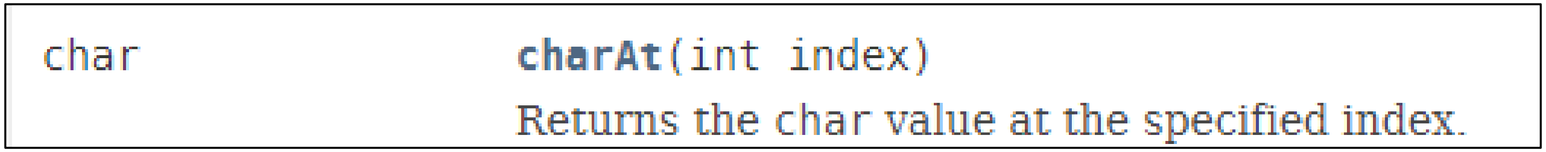
Write your code here!

Hints for Task 3

How to read a character:

0	1	2	3	4
H	E	L	L	O

- String library: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html



char	**charAt**(int index) Returns the char value at the specified index.

```
Scanner in = new Scanner(System.in);
String input = in.next();
char type = input.charAt(0);
            //get the first character
            //from input
```

Lab Exercise Submission

- Submit the following to Moodle
  - ***XXXXXXXX*_ lab02.docx**
  - ***XXXXXXXX*_lab02.zip**

*Replace "***XXXXXXXX***" with your **student ID**

Deadline: Next Monday 11:59 am

References

- Dean, J., & Dean, R. (2008). Introduction to programming with Java: A problem solving approach. Boston: McGraw-Hill.
- Forouzan, B. A., & Gilberg, R. F. (2007). Computer science: A structured programming approach using C (3rd ed.). Boston, MA: Thomson Course Technology.
- Gaddis, T. (2016). Starting out with Java (6th ed.). Pearson.
- Liang, Y. D. (2013). *Introduction to Java programming: Comprehensive version*. (8th ed.). Pearson.
- Schildt, H. (2006). *Java a beginner's guide*. New York: McGraw Hill.
- Wu, C. T. (2010). *An introduction to object-oriented programming with Java*. Boston: McGraw Hill Higher Education
- Xavier, C. (2011). Java programming: A practical approach. New Delhi: Tata McGraw Hill.
- Zakhour, S., Kannan, S., & Gallardo, R. (2013). The Java tutorial: A short course on the basics (5th ed.).
- yet another insignificant Programming Notes. (n.d.). Retrieved from https://www3.ntu.edu.sg/home/ehchua/programming