Lecture Two – Supplement

Introduction to Scanner

Processing input from the console or files

Dr. Martin O’Connor
CA166
www.computing.dcu.ie/~moconnor

Input using the *Scanner* Class

- Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace.

- The resulting tokens may then be converted into values of different types using the various next methods

- Provides methods for reading *byte*, *short*, *int*, *long*, *float*, *double*, and *String* data types from the Java console and other sources.

- *Scanner* is in the *java.util* package and must be imported into your java application.
  - import java.util.Scanner;

A *Scanner* Constructor

Scanner Constructor
`Scanner( InputStream source )` creates a *Scanner* object for reading from *source.* If *source* is *System.in*, this instantiates a *Scanner* object for reading from the Java console (screen)

Example:

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

Scanner next… Methods

Return type	Method name and argument list
`dataType`	`nextDataType( )` returns the next token in the input stream as a *dataType*. The *dataType* can be *byte*, *int*, *short*, *long*, *float*, *double*, or *boolean*
`String`	`next( )` returns the next token in the input stream as a *String*
`String`	`nextLine( )` returns the remainder of the line as a *String*

Prompting the User

- The *next*… methods do not prompt the user for an input value
- Use *System.out.print* to print the prompt, then call the *next*… method.

Example:

```
Scanner scan = new Scanner( System.in );
System.out.print( "Enter your age > " );
int age = scan.nextInt( );
```

SOFTWARE ENGINEERING TIP

End your prompts with an indication that input is expected.

Include a trailing space for readability.

Example:

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

SOFTWARE ENGINEERING TIP

Provide the user with clear prompts for input.

Prompts should use words the user understands and should describe the data requested as well as any restrictions on valid input values.

Example:

```
Enter your first and last name
```

or

```
Enter an integer between 0 and 10
```

Character Input

- *Scanner* does not have a *nextChar* method.
- To read a single character, read the input as a *String*, then extract the first character of the *String* into a *char* variable.

Example:

```
System.out.print( "Enter your " +
                  "middle initial > " );
String initialStr = scan.next();
char middleInitial = initialStr.charAt(0);
```

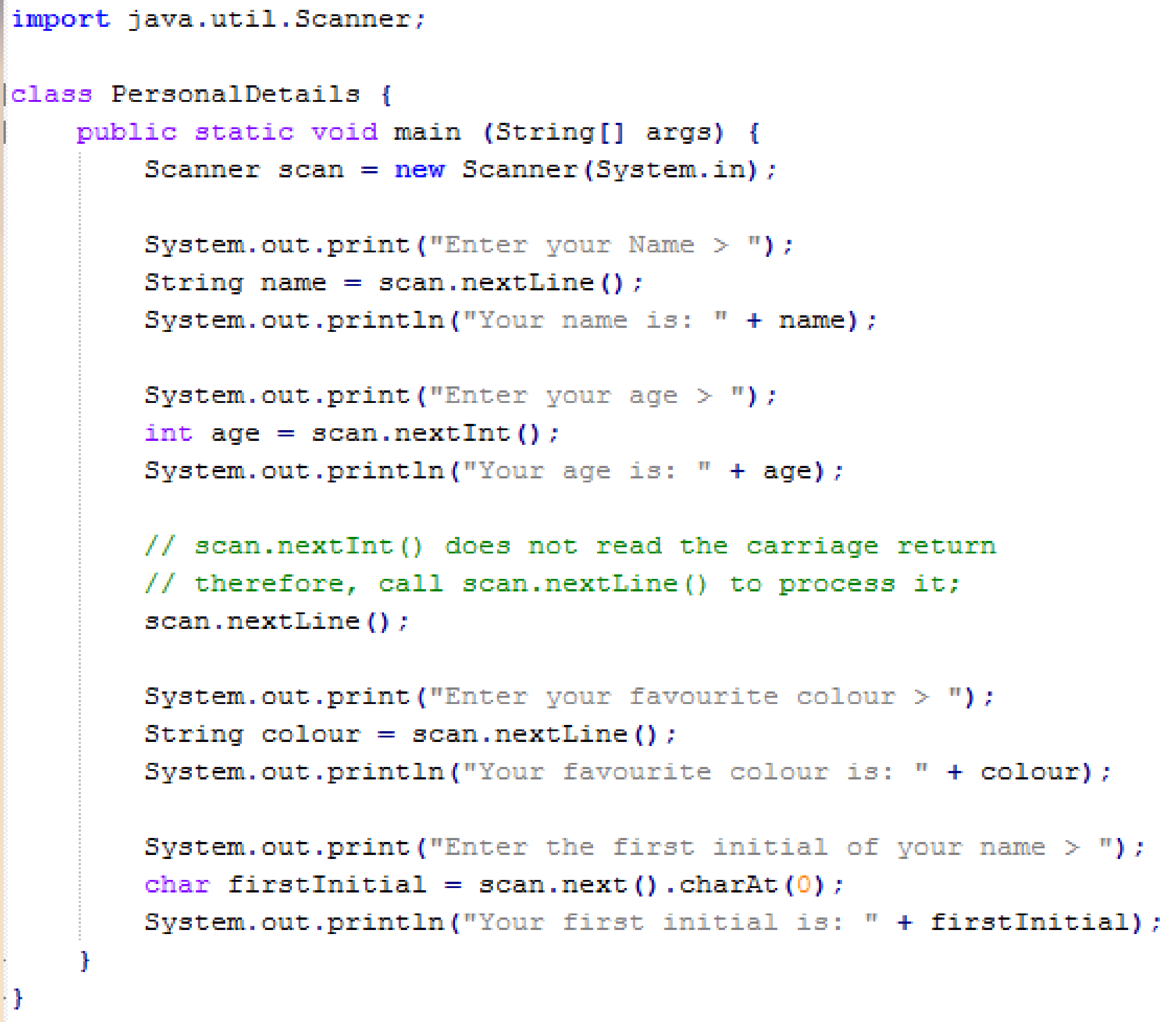
Reading A Whole Line

- The *next* method will read only one word of *String* input because the space is a whitespace character.
- To read a whole line, including spaces, use the *nextLine* method.

Example:

```
System.out.print( "Enter a sentence > " );
String sentence = scan.nextLine();
```

Full Working Example using Scanner



```
import java.util.Scanner;

class PersonalDetails {
    public static void main (String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter your Name > ");
        String name = scan.nextLine();
        System.out.println("Your name is: " + name);

        System.out.print("Enter your age > ");
        int age = scan.nextInt();
        System.out.println("Your age is: " + age);

        // scan.nextInt() does not read the carriage return
        // therefore, call scan.nextLine() to process it;
        scan.nextLine();

        System.out.print("Enter your favourite colour > ");
        String colour = scan.nextLine();
        System.out.println("Your favourite colour is: " + colour);

        System.out.print("Enter the first initial of your name > ");
        char firstInitial = scan.next().charAt(0);
        System.out.println("Your first initial is: " + firstInitial);
    }
}
```