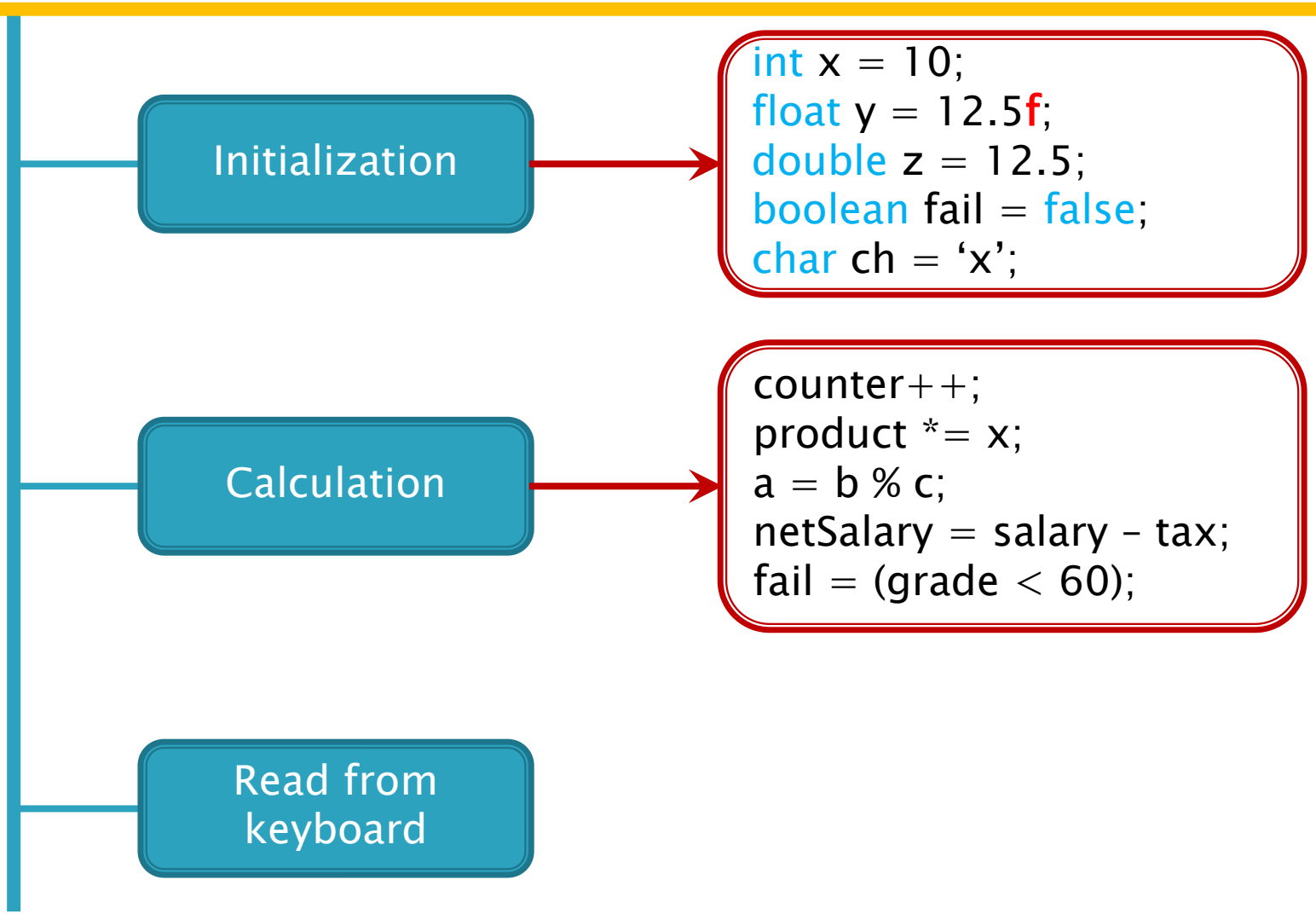
# INPUT STATEMENTS

CSC 111

# Outline

1. Giving values to variables
2. Reading from the keyboard
3. Summary

# 1. GIVING VALUES TO VARIABLES



# Standard Input

- To input primitive data values, we use the Scanner class.
- 4 steps are needed to be able to use input primitive:
  - Step 1: import the Scanner class:
    - `import java.util.Scanner;`
  - Step 2 : declaring a reference variable of a Scanner
    - `Scanner read ;  //we named the object read`
    - **Note : use any name that follow the identifiers rules**
  - Step 3: creating an instance of the Scanner
    - `read = new Scanner (System.in);`
    - **Note : step 2 and 3 can be combined**
  - Step 4: use specific methods to enter data
    - `int x = read.nextInt();`

# 2. READING FROM KEYBOARD

```
1  // import necessary libraries
2  import java.util.*;                  // 1:import the Scanner class
3  public class UserInput1
4  {
5  //2: declaring a reference variable of a Scanner
6                          //3: creating an instance of the Scanner
7     static Scanner console = new Scanner (System.in);
8     public static void main (String[] args)
9       {
10         // Declaration section: to declare needed variables
11             int feet;
12             int inches;
13         // Input section: to enter values of used variables
14             System.out.println ("Enter two integers separated by spaces");
15             feet = console.nextInt(); //4: use specific methods to enter data
16             inches = console.nextInt(); //will read data from keyboared
17         // Processing section: processing statements
18         // Output section: display program output
19             System.out.println ("feet = " + feet);
20             System.out.println ("inches = " + inches);
21      } // end main
   } // end class
```

# 2. READING FROM KEYBOARD

## PROGRAM 1 – ACCEPTING INTEGERS

```
2   import java.util.*;                    //contains the class Scanner
```

- java.util.* is the name of the package (or library) that contains the class Scanner (used in line 5).
- A package is a collection of related classes stored in a file.
- The package should be "imported" in order to use the pre-defined class Scanner.

```
5   // instantiate the object console from the class Scanner
6   static Scanner console = new Scanner (System.in);
```

- Scanner is a Java class defined in the previously imported package java.util.*.
- A class is a non-primitive data type (int, double, etc… are primitive).
- console is the object associated with (of type) Scanner.
- We say that the object console is an instantiation of the class Scanner.
- After this statement, any variable entered through the input device (System.in) uses the object console.

# 2. READING FROM KEYBOARD

## PROGRAM 1 – ACCEPTING INTEGERS

13 `System.out.println ("Enter two integers separated by spaces");`

- "Enter two integers separated by spaces" is displayed to the user.
- The previous string instructs the user with the program requirement.
- This is known as a prompt.

14 `feet = console.nextInt();`

- This statement causes the computer to get the input (the value entered by the user) from the keyboard; and stores it in the variable feet.
- console.nextInt() is a method associated with the Scanner class. It accepts int values.
- If the next input cannot be interpreted as an integer, then the program terminates with a run-time error message indicating an input mismatch.
- Examples of invalid integers include 24w5 or 12.50.

15 `inches = console.nextInt();`

- Same as the statement in line 14. However, the variable used is inches instead of feet.

# 2. READING FROM KEYBOARD

## PROGRAM 1 – ACCEPTING INTEGERS

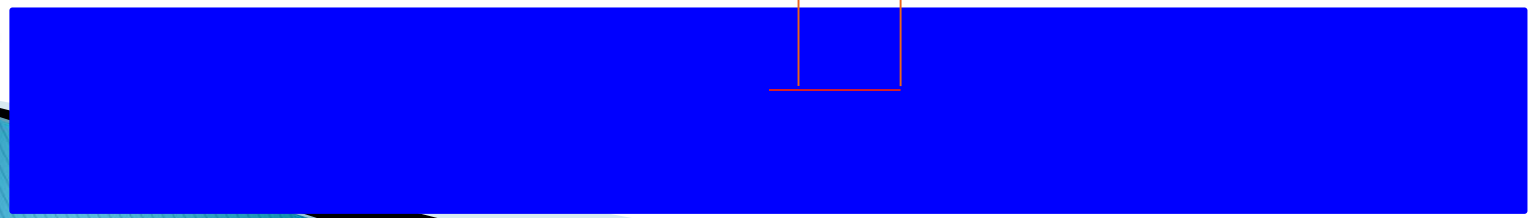
- The output of Program 1 is shown below.
- The numbers in yellow are entered by the user while running the program.

code

```
import java.util.*;
public class UserInput1
{
static Scanner console = new Scanner (System.in);
   public static void main (String[] args)
     {
                int feet;
                int inches;
System.out.println ("Enter two integers separated by spaces");
                feet = console.nextInt();
                inches = console.nextInt();
                System.out.println ("feet = " + feet);
                System.out.println ("inches = " + inches);
    } // end main
 // end class
```

Memory


output



# Common Scanner Methods

| TYPE | METHOD NAME |
| --- | --- |
| Scanner input = new Scanner (System.in); | |
| byte b; | b = input.nextByte(); |
| short sh; | sh = input.nextShort(); |
| long lg; | lg = input.nextLong(); |
| float flt; | flt = input.nextFloat(); |
| char ch; | ch = input.next().charAt(0); |

# 2. READING FROM KEYBOARD

## PROGRAM 2 – ACCEPTING DOUBLE NUMBERS

```
1  // import necessary libraries
2  import java.util.*;                    //contains the class Scanner
3  public class UserInput2
4  {
5     // instantiate the object input from the class Scanner
6     static Scanner input = new Scanner (System.in);
7     public static void main (String[] args)
8       {
9          // Declaration section: to declare needed variables
10            double weight;
11         // Input section: to enter values of used variables
12            System.out.println ("Enter weight>");
13            weight = input.nextDouble();
14         // Processing section: processing statements
15         // Output section: display program output
16            System.out.println ("Weight = " + weight);
17      } // end main
18 } // end class
```

# 2. READING FROM KEYBOARD

## PROGRAM 2 – ACCEPTING DOUBLE NUMBERS

```
13    weight = input.nextDouble();
```

- nextDouble() is a method associated with the class Scanner. It accepts values that can be interpreted as a double type.
  - Integers may be interpreted as a double type.
  - A .0 is added to the integer.
  - If the next input cannot be interpreted as a double, then the program terminates with a run-time error message indicating an input mismatch.
  - Examples of invalid double include 24w5 or 2e10.

# 2. READING FROM KEYBOARD

## PROGRAM 2 – ACCEPTING DOUBLE NUMBERS

- The output of Program 2 is as follows:

```
1  Enter weight >
2  113.6
3  Weight = 113.6
```

- Line 1 prompts the user with the program requirement.
- Line 2 of the above figure represents the user input.
- This is another sample run of Program 2:

```
1  Enter weight >
2  113
3  Weight = 113.0
```

- Note that the user entered an integer number (without decimal point).
- However, nextDouble() interpreted it as a double number.
- Therefore, a .0 is added to the number.

# 2. READING FROM KEYBOARD

## PROGRAM 3 – ACCEPTING STRINGS

```
1  // import necessary libraries
2  import java.util.*;                    //contains the class Scanner
3  public class UserInput3
4  {
5     // instantiate the object read from the class Scanner
6     static Scanner read = new Scanner (System.in);
7     public static void main (String[] args)
8       {
9          // Declaration section: to declare needed variables
10            String firstName, lastName;
11         // Input section: to enter values of used variables
12            System.out.println ("Enter first & last names separated by spaces");
13            firstName = read.next();
14            lastName = read.next();
15         // Processing section: processing statements
16         // Output section: display program output
17            System.out.println ("Name = " + firstName + " " + lastName);
18      } // end main
19 } // end class
```

# 2. READING FROM KEYBOARD

## PROGRAM 3 – ACCEPTING STRINGS

```
13    firstName = read.next();
14    lastName = read.next();
```

- next() is a method associated with the Scanner class. It accepts values that can be interpreted as a String type.
  - Numbers are accepted but interpreted as String type.

# 2. READING FROM KEYBOARD

## PROGRAM 3 – ACCEPTING STRINGS

- The output of Program 3 is as follows:

```
1  Enter first & last names separated by spaces
2  Mariam      A/Azim
3  Name = Mariam A/Azim
```

- Line 1 prompts the user with the program requirement.
- Line 2 of the above figure represents the user input.
- The value of the String ends whenever read.next() encounters a space.
- Note that the first and last names are separated by 5 spaces in the user input.
- However, they are separated by a single space in the output. This is because read.next() ignores spaces entered by the user. However, line 3 of the output obeys the format of the output statement in line 17 of the program.

# 2. READING FROM KEYBOARD

## PROGRAM 4 – ACCEPTING LINES

```
1  // import necessary libraries
2  import java.util.*;                    //contains the class Scanner
3  public class UserInput4
4  {
5     // instantiate the object line from the class Scanner
6     static Scanner line = new Scanner (System.in);
7     public static void main (String[] args)
8       {
9          // Declaration section: to declare needed variables
10            String message;
11         // Input section: to enter values of used variables
12            System.out.println ("Enter a message");
13            message = line.nextLine();
14         // Processing section: processing statements
15         // Output section: display program output
16            System.out.println ("Message = " + message);
17      } // end main
18 } // end class
```

# 2. READING FROM KEYBOARD

## PROGRAM 4 – ACCEPTING LINES

```
13    message = line.nextLine();
```

- nextLine() is a method associated with the Scanner class. It receives the next input as a String until the end of the line.
- nextLine() reads the newline character (Enter), but does not store it as part of the string.
- nextLine() is used when the input data includes spaces. In contrast, next() ignores space characters.

# 2. READING FROM KEYBOARD

## PROGRAM 4 – ACCEPTING LINES

- The output of Program 4 is as follows:

```
1  Enter a message
2  Please follow the instructions written below
3  Message = Please follow the instructions written below
```

- Line 1 prompts the user with the program requirement.
- Line 2 of the above figure represents the user input.
- line.nextLine() stores all line 2 shown in the above figure in the variable message.
- Remember that message is of type String.

# Self-Check Exercises (1)

- Write a program that gets an integer value and prints its double.
- Write a program that gets a string and prints it concatenated with itself.
- Write a program that gets a character value and prints it concatenated with itself.