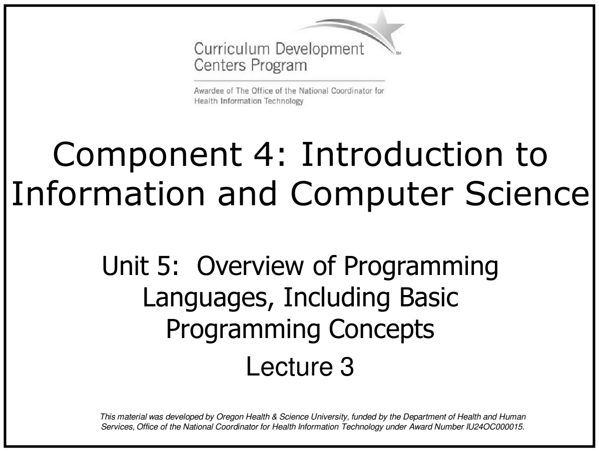


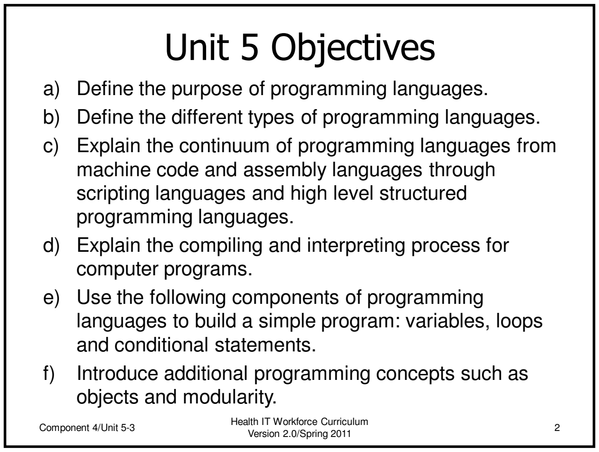
Curriculum Development Centers Program
Awardee of The Office of the National Coordinator for Health Information Technology

Component 4: Introduction to Information and Computer Science

Unit 5: Overview of Programming Languages, Including Basic Programming Concepts

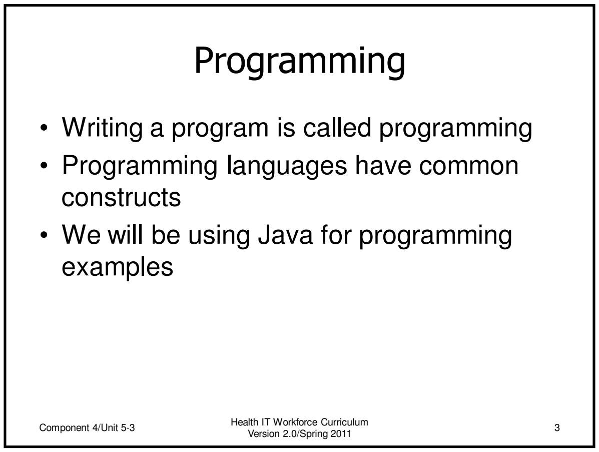
Lecture 3

This material was developed by Oregon Health & Science University, funded by the Department of Health and Human Services, Office of the National Coordinator for Health Information Technology under Award Number IU24OC000015.



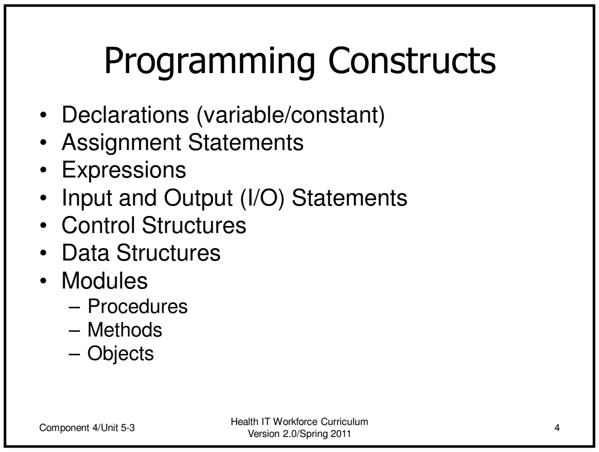
Unit 5 Objectives

a) Define the purpose of programming languages.
b) Define the different types of programming languages.
c) Explain the continuum of programming languages from machine code and assembly languages through scripting languages and high level structured programming languages.
d) Explain the compiling and interpreting process for computer programs.
e) Use the following components of programming languages to build a simple program: variables, loops and conditional statements.
f) Introduce additional programming concepts such as objects and modularity.



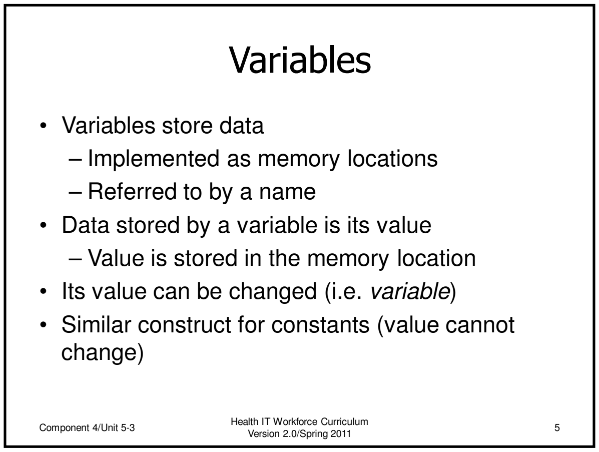
Programming

- Writing a program is called programming
- Programming languages have common constructs
- We will be using Java for programming examples



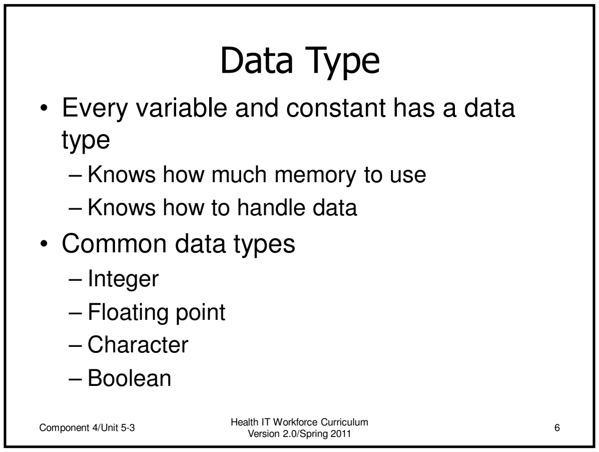
Programming Constructs

- Declarations (variable/constant)
- Assignment Statements
- Expressions
- Input and Output (I/O) Statements
- Control Structures
- Data Structures
- Modules
  - Procedures
  - Methods
  - Objects



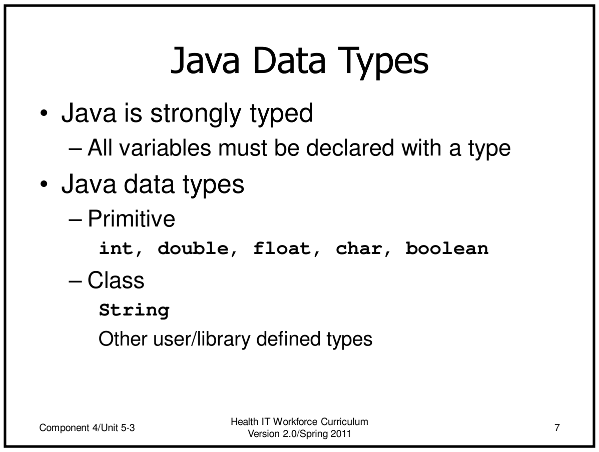
Variables

- Variables store data
  - Implemented as memory locations
  - Referred to by a name
- Data stored by a variable is its value
  - Value is stored in the memory location
- Its value can be changed (i.e. *variable*)
- Similar construct for constants (value cannot change)



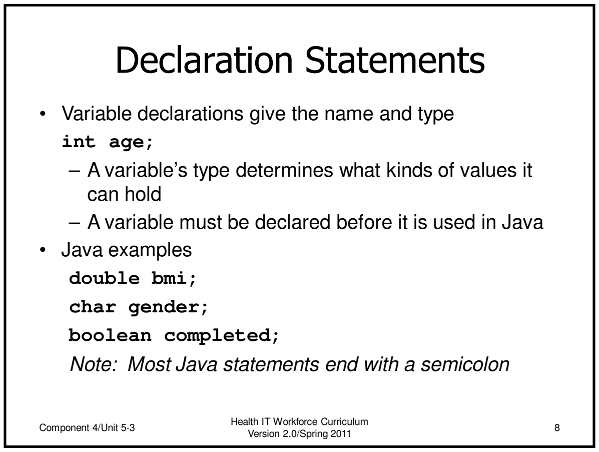
Data Type

- Every variable and constant has a data type
  - Knows how much memory to use
  - Knows how to handle data
- Common data types
  - Integer
  - Floating point
  - Character
  - Boolean



Java Data Types

- Java is strongly typed
  - All variables must be declared with a type
- Java data types
  - Primitive
    `int, double, float, char, boolean`
  - Class
    `String`
    Other user/library defined types

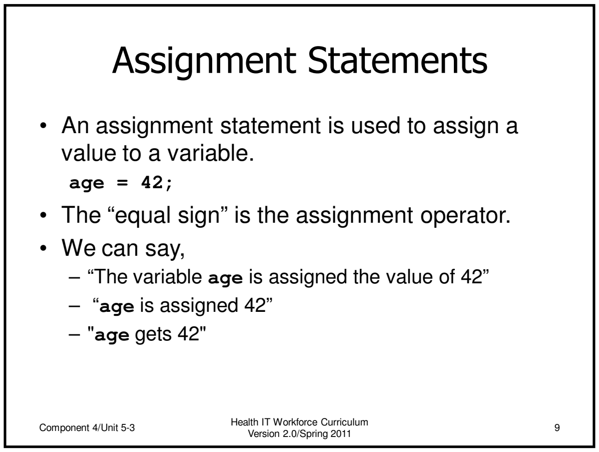


Declaration Statements

- Variable declarations give the name and type
  `int age;`
  - A variable's type determines what kinds of values it can hold
  - A variable must be declared before it is used in Java
- Java examples

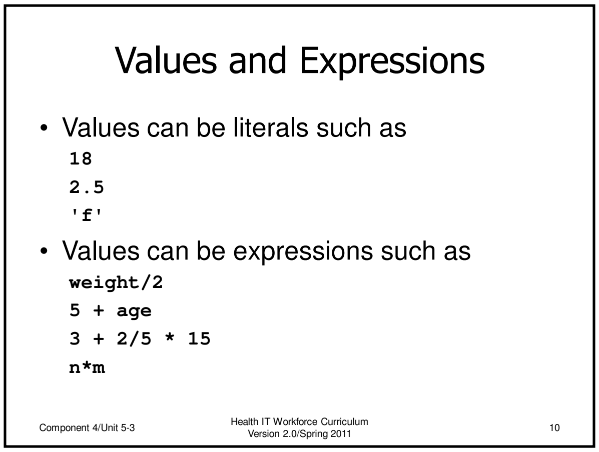
```
double bmi;
char gender;
boolean completed;
```

Note: Most Java statements end with a semicolon



Assignment Statements

- An assignment statement is used to assign a value to a variable.
  `age = 42;`
- The "equal sign" is the assignment operator.
- We can say,
  - "The variable **`age`** is assigned the value of 42"
  - "**`age`** is assigned 42"
  - "**`age`** gets 42"



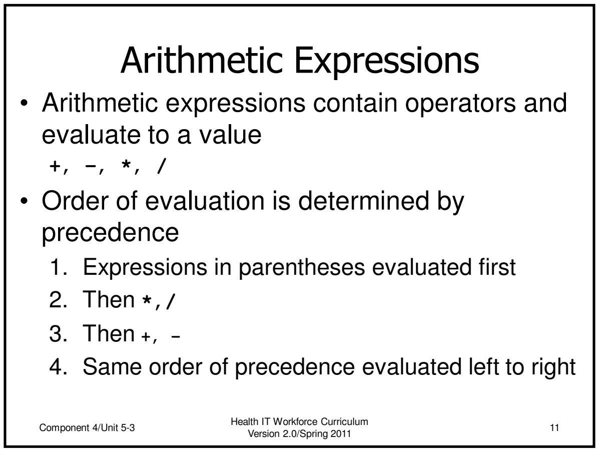
Values and Expressions

- Values can be literals such as

```
18
2.5
'f'
```

- Values can be expressions such as

```
weight/2
5 + age
3 + 2/5 * 15
n*m
```

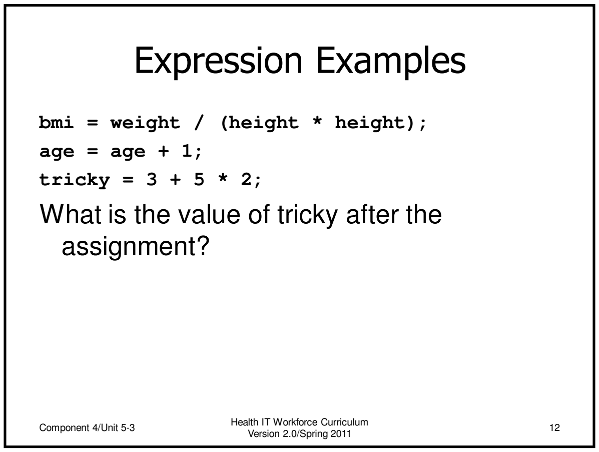


Arithmetic Expressions

- Arithmetic expressions contain operators and evaluate to a value

  `+, -, *, /`

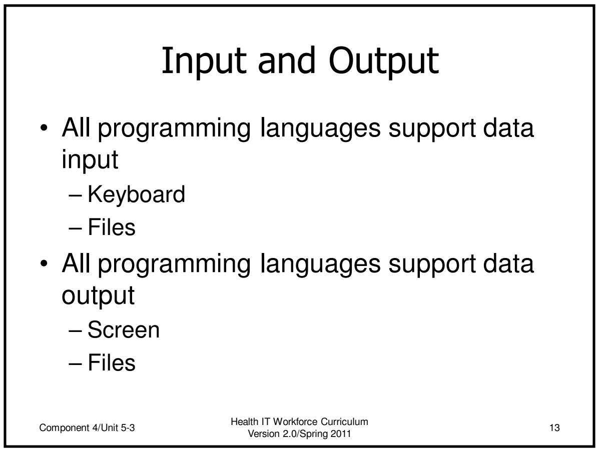
- Order of evaluation is determined by precedence
  1. Expressions in parentheses evaluated first
  2. Then `*,/`
  3. Then `+, -`
  4. Same order of precedence evaluated left to right



Expression Examples

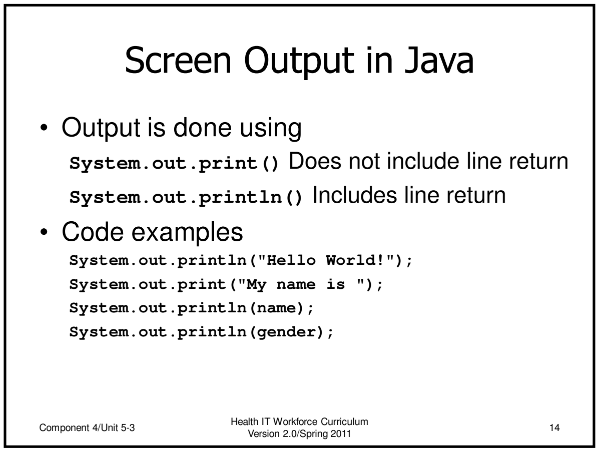
```
bmi = weight / (height * height);
age = age + 1;
tricky = 3 + 5 * 2;
```

What is the value of tricky after the assignment?



Input and Output

- All programming languages support data input
  - Keyboard
  - Files
- All programming languages support data output
  - Screen
  - Files



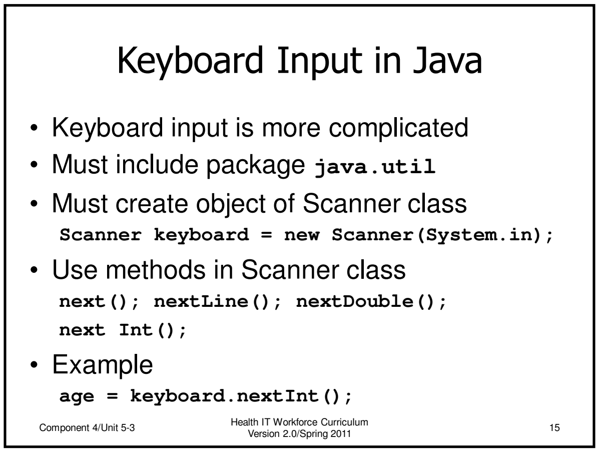
Screen Output in Java

- Output is done using

  `System.out.print()` Does not include line return

  `System.out.println()` Includes line return
- Code examples

```
System.out.println("Hello World!");
System.out.print("My name is ");
System.out.println(name);
System.out.println(gender);
```



Keyboard Input in Java

- Keyboard input is more complicated
- Must include package **`java.util`**
- Must create object of Scanner class

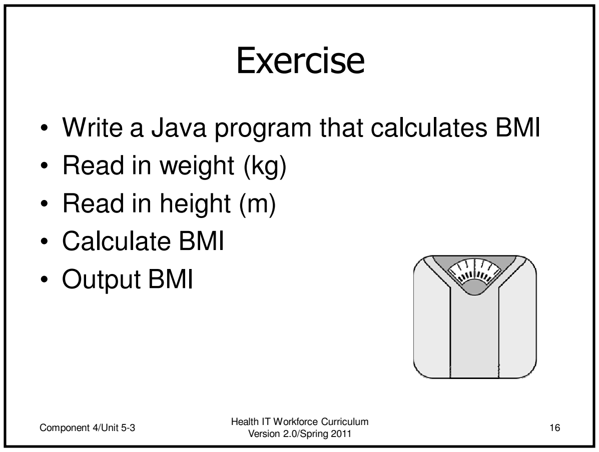
```
Scanner keyboard = new Scanner(System.in);
```

- Use methods in Scanner class

```
next(); nextLine(); nextDouble();
next Int();
```

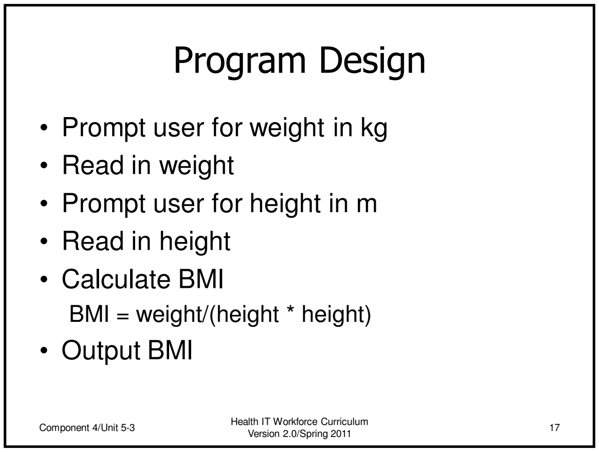
- Example

```
age = keyboard.nextInt();
```



Exercise

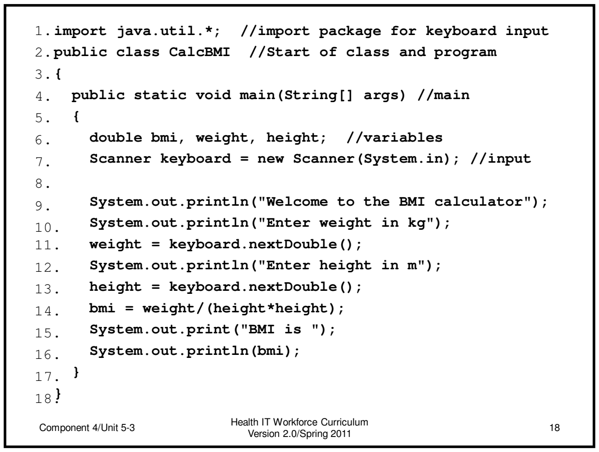
- Write a Java program that calculates BMI
- Read in weight (kg)
- Read in height (m)
- Calculate BMI
- Output BMI



Program Design

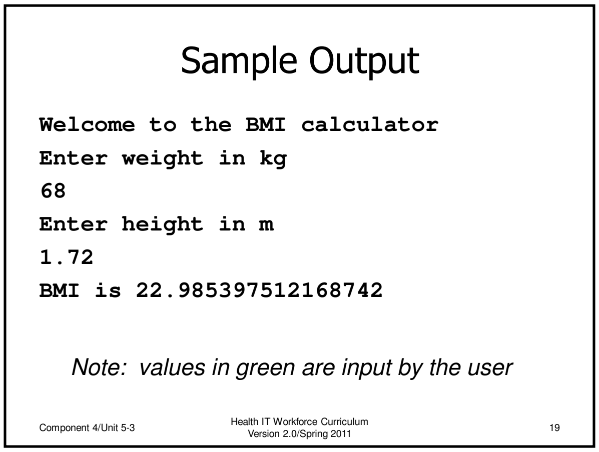
- Prompt user for weight in kg
- Read in weight
- Prompt user for height in m
- Read in height
- Calculate BMI

  BMI = weight/(height * height)
- Output BMI



```
import java.util.*;  //import package for keyboard input
public class CalcBMI  //Start of class and program
{
  public static void main(String[] args) //main
  {
    double bmi, weight, height;  //variables
    Scanner keyboard = new Scanner(System.in); //input

    System.out.println("Welcome to the BMI calculator");
    System.out.println("Enter weight in kg");
    weight = keyboard.nextDouble();
    System.out.println("Enter height in m");
    height = keyboard.nextDouble();
    bmi = weight/(height*height);
    System.out.print("BMI is ");
    System.out.println(bmi);
  }
}
```



Sample Output

```
Welcome to the BMI calculator
Enter weight in kg
68
Enter height in m
1.72
BMI is 22.985397512168742
```

Note: values in green are input by the user