University of York Europe Campus CITY College

Variables and Arithmetic User Input

CCS1110
Programming
Principles and Algorithms

Dr Ioanna Stamatopoulou

Lecture Outline

- Variables
  - Declaration
  - Initialisation
  - Value Assignment
- Primitive Data types
- Arithmetic expressions
- Input

Variables

Variables

- **Data**
  - Information in a form a computer can use/store/process
- A **variable** is memory space used to store data during the execution of a program
- A variable may hold **ONE data value** at a time
  - A number (int, double, float)
  - An alphanumeric (String)
  - A boolean (true or false) value (boolean)
  - A character (char)

Variable Declaration

- To use a variable we must first declare it
  - The **declaration** of a **variable** reserves the appropriate memory on the memory
  - A variable, after it is declared does not contain a value
- To declare a variable, we must declare:
  - The **data type** of the values it will store
  - Its name

Variable Declaration (cont'd)

- The following line of code declares a variable named **`yearOfBirth`** of type **`int`**

```
int yearOfBirth;
```

- The **data type** determines what type of data values can be stored there
  - An **`int`** variable may hold an integer value
  - A **`double`** variable may hold a floating point number value
  - A **`char`** variable may hold one character
- Good practice: use meaningful (long?) variable names that describe the information you are storing there

```
int totalNumberOfCredits;
String name;
```

Variables: Naming conventions

- Variable names must **start** with a **lowercase letter**

```
int sum;
```

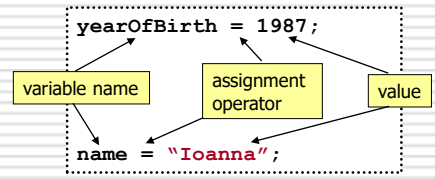
- When the name consists of more than words, we use camelCase for readability:
  - we capitalise the first letter of each word (but the first)

```
String yearOfBirth;
```


Assignment Operator

- To assign (store) a value to a variable we use the **assignment operator =**
  - The assignment operator **=** assigns: the value that appears on its right to the variable that appears on its left

- **CAUTION**: **Assignment is NOT equality**!



Value Assignment

- To the right side of the assignment operator I may use:
  - Concrete values

```
age = 30;
name = "ioanna";
aCharacter = 'c';
```

  - Values stored in variables:

```
x = 5;
y = x; // y will be assigned the value stored in x (5)
```

  - Values that are results of calculations/operations

```
sum1 = 5 + 3;
sum2 = number1 + number2;
// increase by 1:
x = x + 1 ;  // x will become as much as it currently is + 1
```


Declaration and Assignment on one line

- When we know the value of the variable at the moment of declaration, we may declare and assign on one line:

```
int totalNumberOfCredits = 0;
String name = "ioanna";
```

User Input

User input: String

```
import java.util.Scanner;

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

    // Don't forget to prompt the user
    System.out.print("Please enter your name: ");

    // Use the Scanner to read a line of text
    String name = scan.nextLine();

    // Do something with the input string
    // (like, output it to the user)
    System.out.println("Your name is " + name);
  }
}
```

Import the Scanner class

Create a Scanner (only once)

Always prompt the user by printing a **message that explains what they must input** (see next slide)

The method **`nextLine`** of the **`scan`** object returns the sequence of characters (String) that the user enters. This must immediately be stored in an appropriate (String) variable.

Whenever needed, the value that was entered is accessed through the variable name.

User input **Good Practice:**

You must always explain to the user what value you expect them to enter

The user must not wonder or guess what you expect of them

User input: Integers and Doubles

```
import java.util.Scanner;

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

    System.out.print("Please enter your age: " );
    int age = scanner.nextInt();
    System.out.println("You are " + age + " years old");

    System.out.print("Please enter your height (in metres): ");
    double height = scanner.nextDouble();
    System.out.println("You are " + height + " metres tall!");
  } // end of main()

} // end of class HelloWorld
```

Input with a Graphical User Interface (GUI)

GUI Input: String


```
import javax.swing.JOptionPane;
public class StringGUIInput
{
  public static void main(String[] args)
  {
    String name = JOptionPane.showInputDialog("Please, enter your name:");
  }
}
```

The method **`showInputDialog()`** of **`JOptionPane`** displays a graphical input window with whatever message we provide as an argument inside the parentheses, and **always returns a String** with what the user has input.

GUI Input: Numbers

```
public class NumberGUIInput {
  public static void main(String args[]) {
    String ageStr = JOptionPane.showInputDialog("Please, enter your age:");
    int age = Integer.parseInt(ageStr);

    String heightStr = JOptionPane.showInputDialog("Please, enter your height
        in metres:");
    double height = Double.parseDouble(heightStr);
  }
}
```

Because **`showInputDialog()`** always returns a String, if we are asking our user to enter an integer, we must parse the String into an integer with the use of the **`parseInt()`** method of class Integer.

The **`parseInt()`** method receives the String as an argument (inside the parentheses) and returns the integer number inside the String (if one exists).

This integer value we must store in an int variable.

(Similarly if we are asking for a double..)

Input / Output / Variables

```
import java.util.Scanner;

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

      System.out.println("What is your name:")
      name = scan.nextLine();

      System.out.println("What is your age : ");
      age = scan.nextInt();

      System.out.println(name + " you are " + age + " years old.");
   }
}
```

Arithmetic Operators and Expressions

Arithmetic Expressions: Example

```
import java.util.Scanner;
public class Addition
{
   public static void main (String args[])
   {
      Scanner scan = new Scanner(System.in);
      int number1, number2, sum;

      System.out.println("Please, enter the 1st integer number:");
      number1 = scan.nextInt();

      System.out.println("Please, enter the 2nd integer number:");
      number2 = scan.nextInt();

      sum = number1 + number2;
      System.out.println("The sum is " + sum);
   }
}
```

Arithmetic expressions

- Arithmetic operators:
  - Addition `+` `a + b`
  - Subtraction `-` `a - b`
  - Multiplication `*` `a * b`
  - Division `/` `a / b`
  - Modulus `%` `a % b`
    - Modulus returns the remainder of a division
- Examples:
  - `a * (b + c)`
  - `(a + 1) / (b - 3)`
  - `1 + x / (3 * (y + 1) / (z * z) / 3)`

Casting required for non-integer division

- When dividing two integer values Java performs integer division (meaning the result will never be a double value)

```
int result = 10 / 3; //result will get the value 3, not 3.333…
```

- For floating point division, one the operands (numbers must be a floating point

```
double result = 10 / 3.0; //result will get the value 3.333…
double result = 10.0 / 3; //result will get the value 3.333…
```

Casting required for non-integer division (cont'd)

- The techniques used to temporarily convert one data type into another is called **Casting**:

```
int number1 = 10;
int number2 = 3;
double result = (double)number1 / number2;
//result becomes 3.333…
```


Arithmetic expressions: Rules of operator precedence

- Just like in Maths:
  - Parentheses first
    (starting inside going out, from left to right)
  - multiplication, division and modulus next
    (from left to right)
  - addition and subtraction last
    (from left to right)

Input / Output / Variables / Arithmentic

```
import javax.swing.JOptionPane;
public class CalculateUserAge
{
   public static void main (String args[])
   {
      String userYearOfBirth, userName ;
      int yearOfBirth, userAge ;
      userName = JOptionPane.showInputDialog("Your name:");
      userYearOfBirth = JOptionPane.showInputDialog("What is your
                                          year of birth :");
      yearOfBirth = Integer.parseInt(userYearOfBirth);

      userAge = 2004 - yearOfBirth; // Improve using variable

      JOptionPane.showMessageDialog(null, userName + " you are " +
               userAge + " years old.\n", "Calculating Your Age",
               JOptionPane.INFORMATION_MESSAGE);
   } // end of main method
} // end of class
```


Check list

- Do I know what a variable is?
- Can I declare a variable?
- Can I assign a value to a variable?
- Can I get input from the user?
- Can I write a program that performs fundamental arithmetic operations and output the result?
- Consider the operator precedence order and calculate the results of the following expressions:

```
int x = 3 + 5 * ( 7 + 1 ) / 3 + 9 ;
int y = 4 - ( 89 + 3 ) * 7 ;
```