CT60A2411
Java variables, input/output, data types, operators and expression: Week 2

Ashok Kumar Veerasamy, PhD

Learning objectives: This week

- Java variables, constants, methods and classes
- Assignment statements and expressions
- Input and output statements

At the conclusion of this lecutre, students will be able to know defining variables with suitable data types, handling Scanner class for input, naming conventions for variables, methods and classes. In addition, know how to define expressions by using Java's arithmetic operators to formulate expressions for coding.

Recap

1. The comment line in java begins with _____
2. Which of the following are Java keywords/reserved words?

   public, radius, void, class, x, int, salary, String
3. What is main() method in Java?

   a. It is just a subprogram gets executed if it is called like other sub programs in Java.

   b. It is a starting point for JVM (Java virtual machine) to start execution of Java program
4. Why Java is a statically typed language?

   a. Variables must be declared with data type before they used.

   b. All coding lines were written inside main() method only

   c. None of the above
5. Java statement for printing "Welcome to Java" is :

   a. `System.out ("Welcome to Java");`

   b. `System.out.println("Welcome to Java");`

   c. `print("Welcome to Java");`

   d. `System.out.println(Welcome to Java);`

Naming variables and data types for Java

```
public class ComputeArea {
  public static final double PI = 3.14159; //defining constant variable
  /** Main method */
  public static void main(String[] args) {

    // Assign a radius
    float radius = 20;
    double pi = 3.14159;

    // Compute area
    double areaOfCircle = radius * radius * PI;

    // Display results
    System.out.println("The area for the circle of radius " +
      radius + " is " + areaOfCircle);
  }
}
```

What is the difference between *float* and *double* data type?
Similarly, are *byte, short, int, and long* data types represent integer type?
Refer https://www.w3schools.com/java/java_data_types.asp

How to declare *String* type variable? Are *char* (' ') and *String* (" ") same?

How to name variables in Java (identifier)?

- An identifier is a sequence of characters that consist of letters, digits, underscores (_), and dollar signs ($).
- An identifier must start with a letter, an underscore (_), or a dollar sign ($). It cannot start with a digit.
- An identifier cannot be a reserved word.
- An identifier cannot be **true**, **false**, or **null**.
- An identifier can be of any length.

Explain which of the following identifiers are invalid and discuss why are they incorrect?

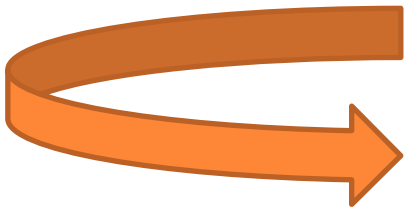
a) A1 b) 1A c) $2_coins d) Do_it e) fish&chips f) got___it

g)_zen h) z00m i) 100_cents j) stop! k) Two numbers l) Seven-11

m) totalNumberOfStudentsEnrolledInThisTutorial n) totStdsInTut

- **The program given below has some errors. Fix it**

```
public class Check1 {  // class name begins with capital letters
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        String 2ndCity; // creating string reference

        byte p = 80000;
        String city = "Lahti"; //non primitive datatype

        System.out.println("Enter another city name");
        2ndCity = sc.next();

        System.out.println("Enter another city population");
        int p1 = sc.nextInt();

        System.out.println(city+" "+"Population is: "+p);
        System.out.println(2ndCity+" +"Population is: "+p1);


    }
}
```



```
--- exec-maven-plugin:3.0.0:exe
Enter another city name
Kouvula
Enter another city population
56000
Lahti Population is: 80000
Kouvula Population is: 56000
------------------------------
```

Java operators and expressions

- 10/3 → 3 10%3 → 1 10.0/3 → 3.33..10/3.0 → 3.33..
- 10.0%3 → 1.0 10.5%3 → ?

```
int x = 5;
x = x++;//postincrement
System.out.print(x);→ 6
```

```
double y = 6.5;
y = --y //predecrement
System.out.println(y);→ 5.5
```

- What is the difference between ++x and x++ / --x and x--?

```
int i = 10;
int newNum = 10 * i++;
```

Same effect as →

```
int newNum = 10 * i;
i = i + 1;
```

```
int i = 10;
int newNum = 10 * (++i);
```

Same effect as →

```
i = i + 1;
int newNum = 10 * i;
```

- **Type casting**

Implicit casting

```
double d = 3; (type widening)
```

Explicit casting

```
int i = (int)3.0; (type narrowing)
int i = (int)3.9; (Fraction part is truncated)
```

What is wrong? int x = 5 / 2.0);

CASTING BETWEEN CHAR AND NUMERIC TYPES

```
int i = 'a'; // Same as int i = (int)'a';
System.out.println(i); → 97

char c = 97; // Same as char c = (char)97;
System.out.println(c); → a
```

- **Self- study: Java expressions**

$$\frac{3+4x}{5} - \frac{10(y-5)(a+b+c)}{x} + 9\left(\frac{4}{x} + \frac{9+x}{y}\right)$$

is translated to

```
(3+4*x)/5 – 10*(y-5)*(a+b+c)/x + 9*(4/x + (9+x)/y)
```

Write Java code that converts a Fahrenheit degree to Celsius using the formula:

$$celsius = \left(\frac{5}{9}\right)(fahrenheit - 32)$$

Self-study

The char type only represents one character. To represent a string of characters, use the data type called String. For example,

```
String message = "Welcome to Java";
```

`String` is actually a predefined class in the Java library just like the `System` class. The `String` type is not a primitive type. It is known as a *reference type*.

```
// Three strings are concatenated
String message = "Welcome " + "to " + "Java";

// String Chapter is concatenated with number 2
String s = "Chapter" + 2; // s becomes Chapter2

// String Supplement is concatenated with character B
String s1 = "Supplement" + 'B';
                          // s becomes SupplementB
```

- **Self-study – Shortcut Assignment Operators**

Operator	*Example*	*Equivalent*
`+=`	`i += 8`	`i = i + 8`
`-=`	`f -= 8.0`	`f = f - 8.0`
`*=`	`i *= 8`	`i = i * 8`
`/=`	`i /= 8`	`i = i / 8`
`%=`	`i %= 8`	`i = i % 8`

Getting input – Scanner class

1. Import Scanner class (put it right after the package statement):

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

2. Create a Scanner object

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

3. Use the methods `next`(), `nextByte`(), `nextShort`(), `nextInt`(), `nextLong`(), `nextFloat`(), `nextDouble`(), or `nextBoolean`() to obtain to a string, byte, short, int, long, float, double, or boolean value.

For example,

```
System.out.print("Enter a double value: ");
Scanner scanner = new Scanner(System.in);
double d = scanner.nextDouble();
```