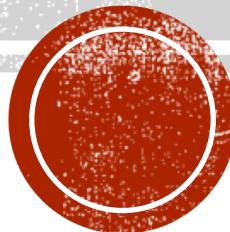




Middlesex  
University  
London

# Lecture 3

# Primitive data types



# REVISION TASK

- Write a Java program which would prompt the user to enter data of a personal details for a survey and print out the information into the console:
- The program must have correct class declaration, main method and using Scanner for input
- The data should include:
  - Name (as a String)
  - Surname (as a String)
  - Gender (as a char, M or F)
  - Print out the data in the output console

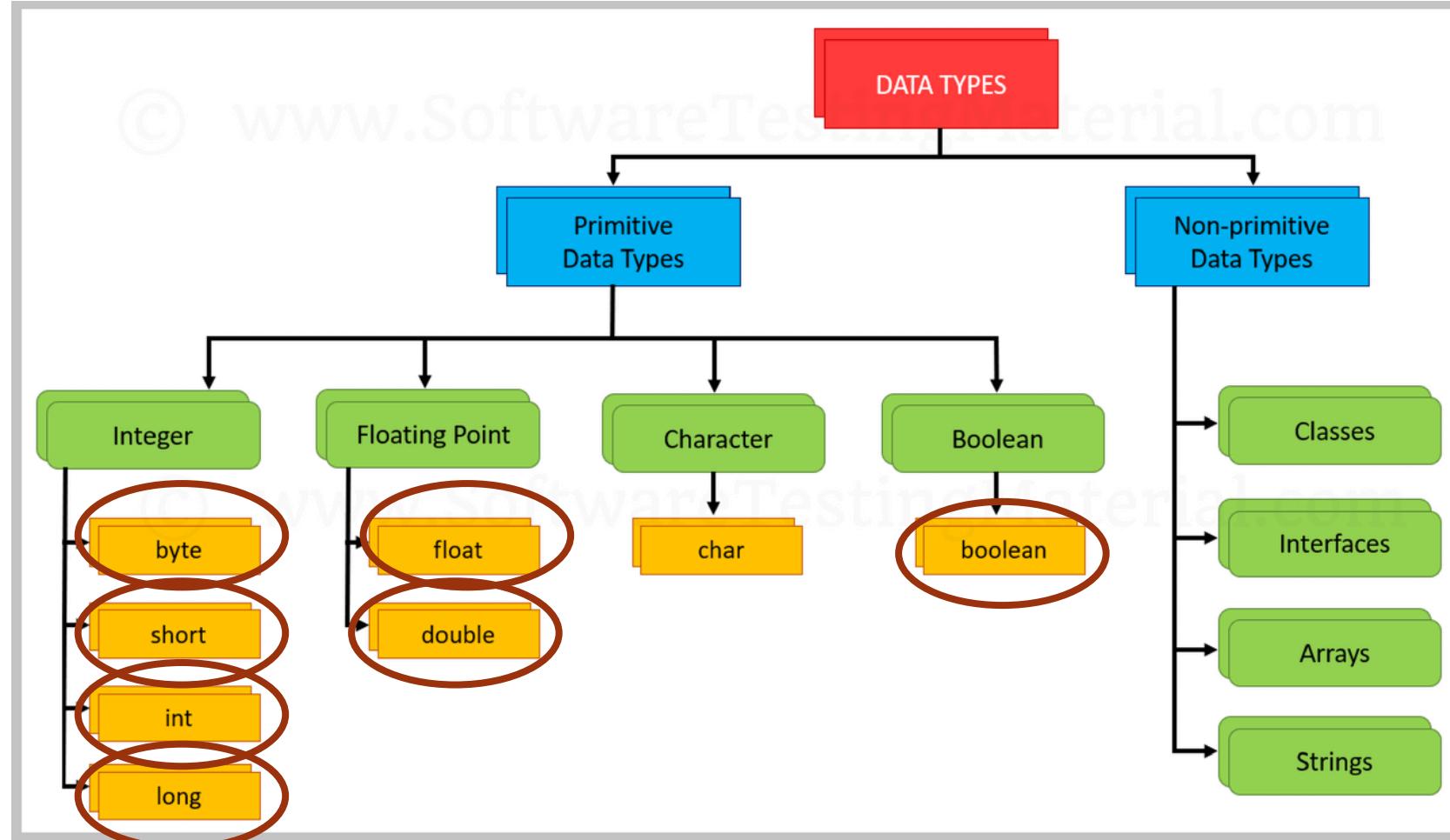


# THIS WEEK

- Primitive data types
- Variables
- Constants
- Operators
- Math class
- Random number generator



# DATA TYPES



# RANGE OF JAVA - PRIMITIVE DATA TYPES

Category	Primitive Data Type	Size	Value (range)
Integer	byte	1 byte	-128 to +127
	short	2 bytes	-32,768 to +32,767
	int	4 bytes	-2,147,483,648 to 2,147,483,647
	long	8 bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Flouting point	float	4 bytes	3.402,823,5 E+38 to 1.4 E-45
	double	8 bytes	1.797,693,134,862,315,7 E+308 to 4.9 E-324
Character	char	2 bytes	Unicode 0 to 65,536
Boolean	boolean	1 bit	true or false



# INTEGER

- Variables declared as integer can hold **whole numbers** such as: -5, -2, 3, 124 etc.
- Integer cannot hold flouting point numbers (numbers with a decimal point)
- *int* is the most commonly used from the four types of Integer data types and mainly used for mathematical calculations
- Example:

```
//integer primitive data types
byte b = 45;
short s = 345;
int i = 533;
long d = 5;
```

```
//alternative way of declaring and assigning value to a variable
int number1 = 100, number2, number3;
number2 = 200;
number3 = number1+number2;

System.out.println(number3);
```



# FLOATING POINT

- Variables declared as floating points can hold **fractional value** such as: -5.2, -2.1, 3.12, 3.14 etc.
- **double** is the most commonly used from the two and mainly used for mathematical calculations
- Example:

```
//floating point primitive data types
float e = (float) 56.4;
double f = 5.1;
```

```
//alternative way of declaring and assigning value to a variable
double number1 = 2.55, number2, number3;
number2 = 6.5;
number3 = number1+number2;
System.out.println(number3);
```



# BOOLEAN

- Variables declared as boolean can hold only true or false .
- Usually used for selection purposes, to either allow or prevent certain action
- Example:

```
// boolean primitive data type
boolean b = true;
boolean a = false;
```



# DATA TYPE CONVERSION

**Java allows for variables to be converted between different types**

- **Implicit conversion** – when the compiler performs a type conversion automatically
  - Example when  $5/2.0$  gives a fractional output
- **Casting** – is forcing conversion of one type to another by prefixing the expression with type
- **Parsing** - refers to the process of determining the syntactic structure of a text

```
//example of Implicit conversion
int number1 =5;
double number1Conv= number1/2.0;
System.out.println(number1Conv);      // output 2.5
System.out.println(number1/2);        // output 2

double numberCasted = number1;
System.out.println(numberCasted);     //outputs 5.0

//example of Casting
double price = 9.99;
int iPrice = (int) price;
System.out.println(iPrice);          //output 9

char at = 64;
int y = (int)(at);
System.out.println(at);             // output @
System.out.println(y);              //output

//example of Parsing
String strNumber = "34";
int iNumber = Integer.parseInt(strNumber);
System.out.println(iNumber +1); //output 35

String doubleNumber = "22.22";
double dNumber = Double.parseDouble(doubleNumber);
System.out.println(dNumber +1); // output 23.22
```



# SCANNER METHODS

```
//importing util package
import java.util.Scanner;

public class test {

    public static void main(String[] args) {

        // creating a Scanner object
        Scanner input| = new Scanner(System.in);

        //used to receive integer variable as input
        input.nextInt();

        //used to receive double variable as input
        input.nextDouble();

        //used to receive Boolean variable as input
        input.nextBoolean();
    }
}
```



# TASK

- Create a new project and declare three variables
  - Double
  - Int
  - Boolean
  - String ( asking user to add number)
- Using scanner to prompt the user for an inputs, assign the values into each type of variable. Convert double into int, String into double and print all to the console.



# CONSTANT

- It's a variable containing a value which does not change for the entire program
- It is considered good practice to minimize the use of variables

```
final dataType VARIABLE_NAME = value;
```

To make a **constant** variable the key word **final** is used

Constant variable name should be all UPPER\_CASE

- Once constant variable is initialized its value cannot be changed



# TASK

- Using your previous project make each variable final and try to run your program again. What was the outcome?



# OPERATORS

Java operators	Meaning	Example	Order of operation
*	Multiplication	$300 * 30 = 9000$	Evaluate <b>first</b> . If there are several operators
/	Division	$1.0 / 2.0 = 0.5$	
%	Reminder	$20 \% 3 = 2$	
+	Addition	$34 + 3 = 37$	Evaluate <b>second</b> . If there are several operators
-	Subtraction	$34.0 - 0.1 = 33.9$	Evaluate <b>last</b>
=	Assignment	<code>a = 34;</code>	



# SHORTCUT ASSIGNMENT OPERATORS

Operator	Example	Equivalent
<code>+=</code>	<code>i += 8</code>	<code>i = i + 8</code>
<code>-=</code>	<code>f -= 8.0</code>	<code>f = f - 8.0</code>
<code>*=</code>	<code>i *= 8</code>	<code>i = i * 8</code>
<code>/=</code>	<code>i /= 8</code>	<code>i = i / 8</code>
<code>%=</code>	<code>i %= 8</code>	<code>i = i % 8</code>



# TASK

Write a program that reads miles in a double value from the console, then converts it in kilometres and displays the result.

The formula for the conversion is as follow  $1 \text{ mile} = 1.6 \text{ kilometres}$

Then do the conversion other way around from kilometers to miles  $1 \text{ kilometre} = 0.62 \text{ miles}$

Can you make any of the values to be final?



# REMAINDER

- Example: If today is Saturday and we want to meet a friend in 10 days time, which day of the week would that be on?

Day 6 in a week is Saturday.

A week in 7 days

$$(6 + 10) \% 7 \text{ is } 2$$

In 10 days time

Day 2 in a week is Tuesday  
Note that day 0 in a week is Sunday



# TASK

- Write a **Java program** which obtains minutes and remaining seconds from an amount of time in second. For example, 500 seconds contains 8 minutes and 20 seconds.
- Using Scanner to prompt the user for an inputs of seconds and calculate the remaining minutes and seconds.



# MATH METHODS

- Java Math class provides several methods to work on math calculations. Example:

Math.pow()	It returns the value of first argument raised to the power to the second argument.
Math.min()	It is used to return the Smallest of two values.
Math.max()	It returns the Largest of the two values.
Math.sin()	It is used to return the trigonometric Sine value of a Given double value.
Math.cos()	It is used to return the trigonometric Cosine value of a Given double value.
Math.ceil()	It is used to find the smallest integer value that is greater than or equal to the argument or mathematical integer.
Math.floor()	It is used to find the largest integer value which is less than or equal to the argument and is equal to the mathematical integer of a double value.
Math.random()	It returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.



# TASKS

- Check the Java API and test few more functionalities from the Math class
- Explore the [Math.random\(\)](#) method. Write each of the statements one by one and write your observations in comments.

```
System.out.println(Math.random());  
  
System.out.println(1 + Math.random());  
  
System.out.println(1 + Math.random() * 10 );  
  
System.out.println((int)(1 + Math.random() * 10 ));
```



# KEY THINGS YOU SHOULD UNDERSTAND AFTER WEEK 3

- Basic variable types (primitives) - identifiers
- Basic assignment with literals and variables
- Number casting
- Reading input from the console with a Scanner
- How to define a constant with the 'final' keyword
- Basic number operations (difference between / and %)
- We are not overly concerned with the Math class – just be aware of what it is and how to reference it in the Java Documentation
  - The point is that you need to learn to navigate the API so that you do not 're-invent the wheel' i.e., always check in case the developers of the Java language have already provided the functionality you need



# MASTERING YOUR SKILLS

- Make sure you finish all your SOBs until today.
- You should be able to do all exercises from Liang book at the end of chapter 2

