جامـــــعـة المـــســتـقـبـل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قـــســـــــــــم الانظمة الطبية الذكية

Lecture: (4)

Basic Computation

Subject: Computer Programming (I)
Level: First
Lecturer: Dr. Maytham N. Meqdad

In Java, data types specify the type of data that a variable can hold. Java has two categories of data types: primitive data types and reference data types.

Primitive Data Types:

1. **Integral Types:**
    - **byte:** 8-bit signed integer. Range: -128 to 127.
    - **short:** 16-bit signed integer. Range: -32,768 to 32,767.
    - **int:** 32-bit signed integer. Range: -2^31 to 2^31 - 1.
    - **long:** 64-bit signed integer. Range: -2^63 to 2^63 - 1.

```
byte myByte = 10;
short myShort = 300;
int myInt = 2000;
long myLong = 123456789L;  // Note: Use 'L' for long literals
```

2. **Floating-Point Types:**

- **float:** 32-bit floating-point. Example: 3.14f.
- **double:** 64-bit floating-point. Example: 3.14.

```
 float myFloat = 3.14f;
double myDouble = 3.14;
```

3.**Character Type:**

char: 16-bit Unicode character.

```
  char myChar = 'A';
```

Boolean Type: boolean: Represents true or false.

```
   boolean myBoolean = true;
```

Reference Data Types:

Object Types:

- Reference data types include objects created from classes. These can be instances of built-in classes or custom user-defined classes.

```
String myString = "Hello, Java!";
```

Arrays:

Arrays are a collection of elements of the same type.

```
int[] myArray = {1, 2, 3, 4, 5};
```

Default Values:

Java assigns default values to variables if no explicit value is provided:

- Numeric types (byte, short, int, long, float, double): 0
- char: '\u0000' (null character)
- boolean: false
- Reference types (objects and arrays): null

Type Casting:

Java supports two types of casting:

Widening (Implicit) Casting:

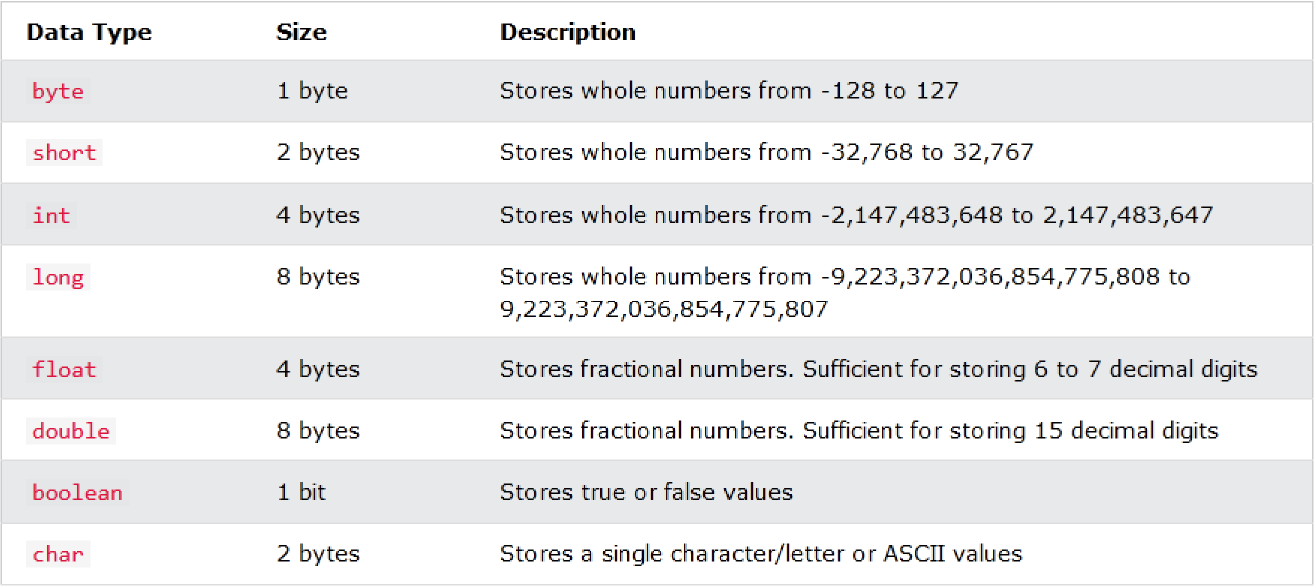
- Automatically done by the compiler.
- No data loss.

```
 int myInt = 100;
long myLong = myInt;  // Widening casting
```

Narrowing (Explicit) Casting:

- Manually performed by the programmer.
- May result in data loss

```
long myLong = 100;
int myInt = (int) myLong;  // Narrowing casting
int myNum = 5;               // Integer (whole number)
float myFloatNum = 5.99f;    // Floating point number
char myLetter = 'D';         // Character
boolean myBool = true;       // Boolean
String myText = "Hello";     // String
```



Data Type	Size	Description
`byte`	1 byte	Stores whole numbers from -128 to 127
`short`	2 bytes	Stores whole numbers from -32,768 to 32,767
`int`	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
`long`	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
`float`	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
`double`	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
`boolean`	1 bit	Stores true or false values
`char`	2 bytes	Stores a single character/letter or ASCII values

Characters

The `char` data type is used to store a **single** character. The character must be surrounded by single quotes, like 'A' or 'c':

```
char myGrade = 'B';
System.out.println(myGrade);
```

Alternatively, if you are familiar with ASCII values, you can use those to display certain characters:

```
char myVar1 = 65, myVar2 = 66, myVar3 = 67;
System.out.println(myVar1);
System.out.println(myVar2);
System.out.println(myVar3);
```

Strings

The `String` data type is used to store a sequence of characters (text). String values must be surrounded by double quotes:

```
String greeting = "Hello World";
System.out.println(greeting);
```

The String type is so much used and integrated in Java, that some call it "the special **ninth** type".

A String in Java is actually a **non-primitive** data type, because it refers to an object. The String object has methods that are used to perform certain operations on strings. **Don't worry if you don't understand the term "object" just yet**. We will learn more about strings and objects in a later chapter.

Java Reserved Keywords

Java has a set of keywords that are reserved words that cannot be used as variables, methods, classes, or any other identifiers:

Keyword	Description
abstract	A non-access modifier. Used for classes and methods: An abstract class cannot be used to create objects (to access it, it must be inherited from another class). An abstract method can only be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from)
assert	For debugging
boolean	A data type that can only store true and false values
break	Breaks out of a loop or a switch block
byte	A data type that can store whole numbers from -128 and 127

case	Marks a block of code in switch statements
catch	Catches exceptions generated by try statements
char	A data type that is used to store a single character
class	Defines a class
continue	Continues to the next iteration of a loop
const	Defines a constant. Not in use - use final instead
default	Specifies the default block of code in a switch statement
do	Used together with while to create a do-while loop
double	A data type that can store whole numbers from 1.7e−308 to 1.7e+308
else	Used in conditional statements
enum	Declares an enumerated (unchangeable) type
exports	Exports a package with a module. New in Java 9
extends	Extends a class (indicates that a class is inherited from another class)
final	A non-access modifier used for classes, attributes and methods, which makes them non-changeable (impossible to inherit or override)
finally	Used with exceptions, a block of code that will be executed no matter if there is an exception or not
float	A data type that can store whole numbers from 3.4e−038 to 3.4e+038
for	Create a for loop
goto	Not in use, and has no function
if	Makes a conditional statement
implements	Implements an interface
import	Used to import a package, class or interface
instanceof	Checks whether an object is an instance of a specific class or an interface
int	A data type that can store whole numbers from -2147483648 to 2147483647
interface	Used to declare a special type of class that only contains abstract methods
long	A data type that can store whole numbers from -9223372036854775808 to 9223372036854775808
module	Declares a module. New in Java 9
native	Specifies that a method is not implemented in the same Java source file (but in another language)
new	Creates new objects
package	Declares a package
private	An access modifier used for attributes, methods and constructors, making them only accessible within the declared class

protected	An access modifier used for attributes, methods and constructors, making them accessible in the same package and subclasses
public	An access modifier used for classes, attributes, methods and constructors, making them accessible by any other class
requires	Specifies required libraries inside a module. New in Java 9
return	Finished the execution of a method, and can be used to return a value from a method
short	A data type that can store whole numbers from -32768 to 32767
static	A non-access modifier used for methods and attributes. Static methods/attributes can be accessed without creating an object of a class
strictfp	Restrict the precision and rounding of floating point calculations
super	Refers to superclass (parent) objects
switch	Selects one of many code blocks to be executed
synchronized	A non-access modifier, which specifies that methods can only be accessed by one thread at a time
this	Refers to the current object in a method or constructor
throw	Creates a custom error
throws	Indicates what exceptions may be thrown by a method
transient	A non-accesss modifier, which specifies that an attribute is not part of an object's persistent state
try	Creates a try...catch statement
var	Declares a variable. New in Java 10
void	Specifies that a method should not have a return value
volatile	Indicates that an attribute is not cached thread-locally, and is always read from the "main memory"
while	Creates a while loop