

# Object Oriented Programming

## Java Programming Language Syntax

Dr. Seán Russell  
`sean.russell@ucd.ie`

School of Computer Science,  
University College Dublin

September XX, 2019

# Table of Contents

- 1 Primitive Data Types in Java
- 2 Data type for text
- 3 Names
- 4 Syntax of Java
  - Statements and expressions
  - Conditional Statements
  - Loops
  - Arrays
- 5 Syntax Errors

# Learning outcomes

After this lecture and the related practical students should...

- be able to recognise the primitive data types in Java
- be able to use the String data type in Java
- understand the rules for identifier names in Java
- be able to explain what an expression is
- be able to determine the type of an expression in Java
- be able to use the basic syntax of Java
- recognise the most common errors shown in Java

# Table of Contents

- 1 Primitive Data Types in Java
- 2 Data type for text
- 3 Names
- 4 Syntax of Java
- 5 Syntax Errors

# Data types in C and Java

- Most primitive types are the same
- Sometimes different size
- There are some new data types

## Structs

Structs in C are replaced by classes and objects. But classes and objects are much more powerful.

# Primitive data types

- Two types of data
- Others are based on these
  - ▶ Integers
  - ▶ Floating point numbers

# Integer Types

Name in C	Size in C	Name in Java	Size in Java
char	== 1 byte	char	== 2 bytes
		byte	== 1 byte
int	>= 2 bytes	short	== 2 bytes
long	>= 4 bytes	int	== 4 bytes
long long	>= 8 bytes	long	== 8 byte

## Examples

```
int age = 35;  
long count = 12345L;  
byte c = 100;  
char letter = 'a';
```

# Floating Point Types

Name in C	Size in C	Name in Java	Size in Java
float	== 4 byte	float	== 4 bytes
double	== 8 bytes	double	== 8 bytes

## Examples

```
float interest = 2.5f;  
double max = 543.45;
```



# Logical Type

Java uses the **boolean** type for logic conditions

## boolean

Boolean can contain only two values: **true** or **false**.

- boolean is the type of a comparison such as `x == 0`
- Uses the same conditional operators as C. e.g. `||` (OR) + `&&` (AND) + `!` (NOT)

## Examples

```
boolean an = false;  
boolean sw = x > 45 && x < 100;
```

# Operators for Numbers

- **x + y** Add two values  
`int r = 1 + 4;`
- **x - y** Subtract one value from another  
`int r = 6 - 2;`
- **a \* b** Multiply two values  
`double ans = 4.5 * 4;`
- **a / b** Divide one value by another  
`double ans = 4.5 / 4;`
- **a % b** Modulus operator, calculates the remainder of the integer division `a \ b`  
`int ans = 1234 % 4;`

# Operators for Numbers

- **x + y** Add two values  
`int r = a + b;`
- **x - y** Subtract one value from another  
`int r = a - b;`
- **a \* b** Multiply two values  
`double ans = a * b;`
- **a / b** Divide one value by another  
`double ans = a / b;`
- **a % b** Modulus operator, calculates the remainder of the integer division  $a \setminus b$   
`int ans = a % b;`

# Operators for Booleans

- **x || y** OR operator

```
boolean r = a || b;
```

```
boolean r = x < 0 || x > 100;
```

r is true if either a **or** b are true

# Operators for Booleans

- **x && y** AND operator

```
boolean r = a && b;
```

```
boolean r = x > 0 && x < 100;
```

r is true if both a **and** b are true

# Operators for Booleans

- **!****a** NOT operator

```
boolean r = !a;
```

r is true if a is false

```
boolean r = !(x > 0) ;
```

r is true if x is NOT greater than 0

# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

5 Syntax Errors

# Type for text

In C if we wanted to store a sequence of characters we would use an array of chars, in Java we use **String**

- String is a class
- It is important to remember that String starts with a **uppercase S**

## Examples

```
String name = "Sean";  
String message = "hello world!";
```



# Operator for text

There is only one basic operator for Strings. This is the **concatenation** operator which adds strings together.

- **x + y** Adds two strings together to create a new String

```
String r = "hello " + "world";  
String s = r + "!";
```

# Operator for text

## String doesn't change

It is important to note that the concatenation operator does not change the values of Strings. It **creates a new String** and leaves the original the same.

In our example, the variable `r` will contain `"hello world"` and `s` will contain `"hello world!"`

# Concatenation with Numbers

## Example

```
String r = "Hello, that will cost " + 5;  
r → "Hello, that will cost 5"
```

We can also use the concatenation operator many times in the same statement

## Example

```
String s = "Hello, that will cost " + 5 + "  
RMB";  
s → "Hello, that will cost 5 RMB"
```

# Data types example

```
1 public class Variables {  
2     public static void main(String[]  
3         args) {  
4         int x = 5;  
5         int y = 6;  
6         int z = x + y;  
7         String m = x + "+" + y + "=" + z;  
8         System.out.println(m);  
9         boolean b = true;  
10        System.out.println("b = " + b );  
11    }  
}
```

# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

5 Syntax Errors

# Identifiers

The name we give something in Java is an **identifier**

There are many rules for identifiers:

- They can contain letters, digits and underscore (\_)
- They cannot start with a digit
- They cannot contain symbols
- They cannot contain spaces
- They cannot be a reserved word
- They are **case sensitive**

# Conventions for identifiers

- Identifiers **should** be descriptive
- Variable names **should** start with a lowercase letter
- Class names **should** start with an uppercase letter
- > one word should use camel case

## Examples

```
int height = 178;  
int boxWidth = 40;
```

# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

- Statements and expressions
- Conditional Statements
- Loops
- Arrays

5 Syntax Errors



# Syntax of Java

- **Syntax** is a word used in programming to describe the **grammar** of a programming language.
- Computers are stupid!!
- Grammar must be **perfect**
- Grammatical errors in programming are called **syntax errors**

# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

- Statements and expressions
- Conditional Statements
- Loops
- Arrays

5 Syntax Errors

# Statements and Expressions

- A **statement** is a piece of code that does something
- An expression is a piece of code that can be calculated to give us a **result**
- Statements can contain expressions

# Statements

Usually have a semi-colon (;) at the end

## Examples of statements

- **Variable declaration**

**Syntax:** type name; e.g. `int number;`

- **Variable assignment**

**Syntax:** name = expression; e.g. `number = 7;`

- **Calling a method**

**Syntax:** `object.methodName(parameters);`  
e.g. `System.out.println("Hello");`

# Expressions

A piece of code that can give us an **answer**

- These may include operators
- Expressions are usually part of a statement

## Examples of expressions

```
1  
number  
number + 2  
"Hello, my name is " + name
```

Expressions have a **type**

# Conditional expressions

A conditional expression has the type boolean

## Examples of conditional expressions

```
true
```

```
x < 100
```

```
x < 100 && x > 0
```

# Calculating expression types

There is logic to calculating the type of an expression

- If all parts are the same data type, then that is the type of the expression  
e.g. `7 + 7 + 67 + 9`
- If all parts are the same type, but not data type, then it is **largest type**  
e.g. `int + short + byte + long`
- If the components are different types, integers and floating point numbers and strings, then ask yourself can I save this as a X for each type  
e.g. `int + double`

# Expressions

## Questions

For each of the following expressions, what is the correct type?

① `12 + 6.7`

② `500 / 10`

③ `5 * 4.5f`

④ `"That will cost you " + 1 + 10 + " RMB"`

⑤ `123453L / 1234`

⑥ `100 < 56`

⑦ `false || ( true && false)`



# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

- Statements and expressions
- **Conditional Statements**
- Loops
- Arrays

5 Syntax Errors

# The if statement

In Java the condition must of an if statement must be an boolean expression

## Syntax

```
1 if(booleanExpression){  
2     code  
3 }
```

## Example

```
1 if(x > 5){  
2     System.out.println("hello");  
3 }
```

# The if-else statement

## Syntax

```
1 if(booleanExpression){  
2     code  
3 } else {  
4     other code  
5 }
```

## Example

```
1 if(booleanExpression){  
2     System.out.println("hello");  
3 } else {  
4     System.out.println("goodbye");  
5 }
```

# Switch Statement

- Like a series of if-else statements
- Pass an expression to the statement and it will choose the correct case to **begin** executing
- All code after the selected case will be executed
- Can use an integer, Strings or enumerated type

# Syntax of Switch

```
1 switch(expression) {  
2     case constant-expression:  
3         statement(s);  
4         break;  
5     case constant-expression:  
6         statement(s);  
7         break;  
8     default:  
9         statement(s);  
10 }
```

# Example of Switch

```
1 switch(numValues){  
2 case 1:  
3     doSomething();  
4     break;  
5 case 5:  
6     doSomethingElse();  
7     break;  
8 default:  
9     doSomethingDifferent();  
10 }
```

# Example of Switch

```
1 switch(numValues){  
2   case 1:  
3     doSomething();  
4   case 5:  
5     doSomethingElse();  
6   default:  
7     doSomethingDifferent();  
8 }
```

# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

- Statements and expressions
- Conditional Statements
- Loops
- Arrays

5 Syntax Errors



# Syntax of Java

## The while loop

### Syntax

```
1 while(booleanCondition){  
2     repeat some code  
3 }
```

### Example

```
1 while(x > 5){  
2     System.out.println("hi");  
3     x++;  
4 }
```

# Syntax of Java

## The for loop

### Syntax

```
1 for(initialise variable; booleanExpression;  
   counter){  
2     repeat some code  
3 }
```

### Example

```
1 for(int i = 0; i < 5; i++){  
2     System.out.println("Hi");  
3 }
```

# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

- Statements and expressions
- Conditional Statements
- Loops
- Arrays

5 Syntax Errors

# Arrays

- Arrays in Java are be a little different
- Array declarations are different
- Arrays store more information
- Arrays are objects

# Declaring an Array

- In C, `int arrayName[arraySize];` declares a variable called `arrayName` and also **allocates memory** for numbers
- In Java, `int[] arrayName;` only declares a variable called `arrayName`

# Creating an Array

- In Java we must both declare and construct an array
- Construction of an array uses the syntax  
`new type[arraySize]`
- `type[] arrayName = new type[size];`
- `int[] numbers = new int[10];`

# Using an Array

- Using an array is the same
- To put 123 in element 0 of an array called numbers, then the statement is `numbers[0] = 123;`
- Accessing the value in element 7 of the array called numbers we use the expression `numbers[7]`

# Array Size

- Arrays in Java know their size
- The size of an array is stored in a special instance variable called `length`
- To find out the size of an array we use `arrayName.length`
- For example, `numbers.length` would give us the value 10



# Arrays and Loops

- A loop to visit every element in an array can be written based on a template
- The following code can be used to loop through any array no matter what size, simply by replacing aN with the name of you array

```
for(int i = 0; i < aN.length; i++)
```

# Table of Contents

1 Primitive Data Types in Java

2 Data type for text

3 Names

4 Syntax of Java

5 Syntax Errors

# Syntax errors

- Syntax errors are shown whenever we try to **compile** a Java file
- Remember any syntax errors so you can solve them next time

## One error at a time

- Many error messages are not easy to understand
- Many errors may mean a single mistake
- Always fix errors **one at a time** and compile again

# Syntax errors

## Typical syntax errors

Here is a list of some of the most common syntax errors, we will have a look at the results of these.

- File name and class name **do not match**
- Misspelled a keyword
- Misspelled a variable name
- Incorrect variable name
- Forgotten semicolon
- Forgot to import a library

# Syntax Errors

File name and class name **do not match**

```
1 public class Hello {  
2     public static void main(String[] args){  
3         System.out.println("Hello, Sean");  
4     }  
5 }
```

```
1 FName.java:3: error: class Hello is public,  
    should be declared in a file named  
    Hello.java  
2 public class Hello {  
3         ^  
4 1 error
```

# Syntax Errors

Misspelled a keyword

```
1 public Class MSpell{  
2     public static void main(String[] args){  
3         System.out.println("Hello, Sean");  
4     }  
5 }
```

```
1 MSpell.java:3: error: class, interface, or enum expected  
2 public Class MSpell{  
3     ^  
4 MSpell.java:4: error: class, interface, or enum expected  
5 public static void main(String[] args){  
6     ^  
7 MSpell.java:6: error: class, interface, or enum expected  
8 }  
9 ^  
10 3 errors
```

# Syntax Errors

Misspelled a variable name

```
1 public class MSpell2{
2     public static void main(String[] args){
3         String varName = "Sean";
4         System.out.println("Hello, " + varname);
5     }
6 }
```

```
1 MSpell2.java:6: error: cannot find symbol
2 System.out.println("Hello, " + varname);
3                                     ^
4 symbol: variable varname
5 location: class MSpell2
6 1 error
```

# Syntax Errors

## Incorrect variable name

```
1 public class MSpell3{
2     public static void main(String[] args){
3         int 2cool = 34;
4         System.out.println("Hello, " + 2cool);
5     }
6 }
```

```
1 MSpell3.java:5: error: not a statement
2     int 2cool = 34;
3     ^
4 MSpell3.java:5: error: ';' expected
5     int 2cool = 34;
6     ^
7 MSpell3.java:6: error: ')' expected
8     System.out.println("Hello, " + 2cool);
9                     ^
10 MSpell3.java:6: error: illegal start of expression
11     System.out.println("Hello, " + 2cool);
12                     ^
13 4 errors
```



# Syntax Errors

## Forgotten semicolon

```
1 public class ForgetS{  
2     public static void main(String[] args){  
3         String name = "Sean"  
4         System.out.println("Hello, " + name);  
5     }  
6 }
```

```
1 ForgetS.java:5: error: ';' expected  
2     String name = "Sean"  
3                     ^  
4 1 error
```

# Syntax Errors

Forgot to import a library

```
1 public class ForgetI{
2     public static void main(String[] args){
3         Scanner in = new Scanner(System.in);
4         String name = "Sean";
5         System.out.println("Hello, " + name);
6     }
7 }
```

```
1 ForgetI.java:5: error: cannot find symbol
2     Scanner in = new Scanner(System.in);
3     ^
4     symbol: class Scanner
5     location: class ForgetI
6 ForgetI.java:5: error: cannot find symbol
7     Scanner in = new Scanner(System.in);
8     ^
9     symbol: class Scanner
10    location: class ForgetI 2 errors
```