Java
Introduction
Data types and operators

Computer programming development

- Connect wires to program the hardware
- Enter binary numbers by hand (machine code)
- Assembler program created: human oriented coding
- “High level” languages created
  Fortran, Cobol, Basic
- Structured programming languages
  Pascal, C
- Object-oriented programming languages & techniques
  SmallTalk, C++, Java

Object oriented fundamentals

- Programs organized around data

- Encapsulation
  - Data and code wrapped together = > Object
  - Data and code can be private to their object
  - Some code public
  - Data can be made public

Classes

- Class is the basic unit of encapsulation
- It specifies the data & code to operate on the data

- Objects are instances of a class

- Code and data that make up a class are considered members of the class
    - member variables = the data
    - code operating on data => member methods

- Method is Java for a subroutine (assembly)
- Method is Java for a function (from C/C++)

Java Identifiers (names: variables, methods, etc)

- Java is case sensitive
- First letter must be: alpha, underscore, dollar sign
- Don’t use standard method names for your identifiers
- Don’t use Java keywords for your identifiers

abstract	assert	boolean	break	byte	case
catch	char	class	const	continue	default
do	double	else	enum	extends	final
finally	float	for	goto	if	implements
import	instanceof	int	interface	long	native
new	package	private	protected	public	return
short	static	strictfp	super	switch	synchronized
this	throw	throws	transient	try	void
volatile	while				

Table 1-1 The Java Keywords

Type	Meaning
boolean	Represents true/false values
byte	8-bit integer
char	Character
double	Double-precision floating point
float	Single-precision floating point
int	Integer
long	Long integer
short	Short integer

16

64

32

Table 2-1 Java's Built-in Primitive Data Types

Integer types

Type	Width in Bits
byte	8
short	16
int	32
long	64

print() println()

System.out.println(“Hello \t World”);

Escape Sequence	Description
\'	Single quote
\"	Double quote
\\	Backslash
\r	Carriage return
\n	New line
\f	Form feed
\t	Horizontal tab
\b	Backspace
ddd	Octal constant (where *ddd* is an octal constant)
\u*xxxx*	Hexadecimal constant (where *xxxx* is a hexadecimal constant)

Table 2-2 Character Escape Sequences

Java defines the following arithmetic operators:

Operator	Meaning
+	Addition (also unary plus)
–	Subtraction (also unary minus)
*	Multiplication
/	Division
%	Modulus
++	Increment
– –	Decrement

The relational operators are shown here:

Operator	Meaning
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

The logical operators are shown next:

Operator	Meaning
&	AND
|	OR
^	XOR (exclusive OR)
||	Short-circuit OR
&&	Short-circuit AND
!	NOT

The outcome of the relational and logical operators is a **boolean** value.

For the logical operators, the operands must be of type **boolean**, and the result of a logical operation is of type **boolean**. The logical operators, **&**, **|**, **^**, and **!**, support the basic logical operations AND, OR, XOR, and NOT, according to the following truth table:

p	q	p & q	p | q	p ^ q	!p
False	False	False	False	False	True
True	False	False	True	True	False
False	True	False	True	True	True
True	True	True	True	False	False

Highest						
++ (postfix)	– – (postfix)					
++ (prefix)	– – (prefix)	~	!	+ (unary)	– (unary)	*(type-cast)*
*	/	%				
+	–					
>>	>>>	<<				
>	>=	<	<=	instanceof		
==	!=					
&						
^						
|						
&&						
||						
?:						
->						
=	op=					
Lowest						

Table 2-3 The Precedence of the Java Operators