ARCTICTIGER

Chapter 1
Java Fundamentals

Based on the course literature:
Java: A beginner's guide
Fifth Edition
Herbert Schildt

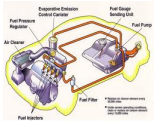
What we'll cover

- History of Java
- Bytecode
- Create, compile and run in Java
- Variables
- if and for control statements

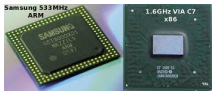
ARCTICTIGER

History of Java



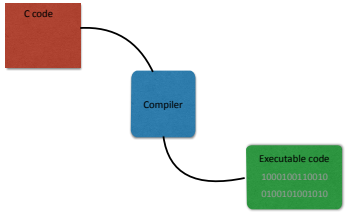
ARCTICTIGER



ARCTICTIGER

Bytecode

ARCTICTIGER 5

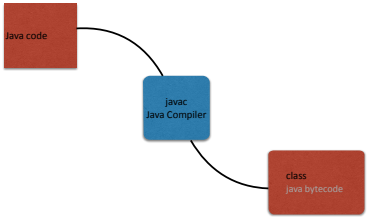


ARCTICTIGER 6

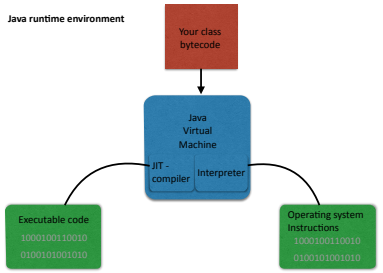
Write Once Run Anywhere

Write Once Debug Everywhere

ARCTICTIGER 7



ARCTICTIGER 8



ARCTICTIGER
9

- As long as the Java Runtime Environment is installed on a system, a program can run.
- The programmer doesn't have to think about the chipset.
- Since the JVM runs the code it can place certain security restrictions on the code.
- It also allows the program to be dynamic allowing objects to be allocated at runtime.

ARCTICTIGER
10

Create, compile and run

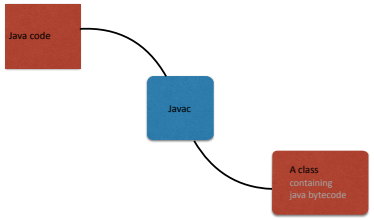
ARCTICTIGER
11

Javac

The JDK

- The Java Developer Kit (JDK) is required to compile our code to bytecode that can run on the Java Runtime Environment.
- The JDK includes java compiler (javac) command line program that performs this task
- Download and install the latest jdk from oracle.com

ARCTICTIGER
12



ARCTICTIGER 13

Important

- Your java file should have the same name as the class it contains.
- Classes should always start with a Capital letter and therefore so should your filename.

ARCTICTIGER 14

Demo

- Create
- Compile
- Run

ARCTICTIGER 15

Comments

```
/*
    This
    is
    a
    multiline
    comment
 */

// Single line comments look like this
```

ARCTICTIGER 16

Class

```
class HelloWorld {}
```

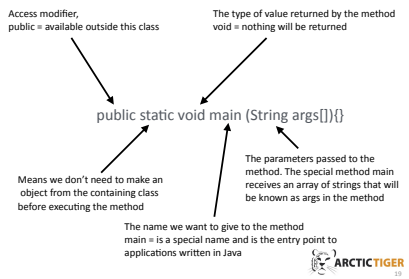
A class is Java's basic unit of encapsulation. All Java programs consist of one or more class.

All program activity exists in a class.

Method

```
public static void main (String args[]){
}
```

- A method/function/subroutine is declared as above.
- The method name main is special and indicates to our compiler that this is the entry point into our code.
- All code that should be executed when a method is called should be within the {} block.



Variables

- Variable : a named memory location that can be assigned a value.
- **Java is strongly typed**. This means variable type must be specified when declaring variables.
- The main advantage of this is Java can allocate just the required amount of memory required to store the variable type.
- Code can executed quicker. E.g. integer calculations are faster than floating point calculations.
- It is also easier to read a program when you can see the variable types.
- It makes writing coding tools such as "code completion" simpler.

```
int var1;
```

The above code declares an integer variable called var1 and allocates enough memory to store any integer.

```
var1 = 1024;
```

The above code fills the variable var1 memory with the integer value of 1024

```
int var1 = 1024;
```

Declaration and assignment can be done in one line too.

Naming rules

The first letter in an identifier can only be $ _ or a letter. Numbers are not permitted.

```
int 12x; // This is not allowed
int x12; // This is OK.
```


Naming conventions

- Classes - CamelCase e.g. class Customer
- Methods - mixedCase e.g. void calculateTax()
- Variables - mixedCase e.g. string firstName

We'll come back to this as we progress.

http://java.about.com/od/javasyntax/a/nameconventions.htm

Java Keywords

- http://en.wikipedia.org/wiki/List_of_Java_keywords

Exercise

Write a Program that converts gallons to liters.

1) Create a new file GalToLit.java

2) Use 2 variables of type double one called gallons one called liters.

3) Set the value of gallons to 10.

There are 37.854 liters in a US gallon

4) Set the value of liters to be the result of the above calculation.

5) Write the result out to the terminal.

if and for control statements

The if statement

- Hint it's the same as JS
- Unlike JS though Java is strongly typed this means there is never a problem with type coercion so == in Java is the same as === in JS.

```
if (condition) statement;
else statement;
```


Basic condition operators

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

The for loop

```
for (initialization; condition; iteration) statement;
for(int i = 0; i < 10; i++){
        System.out.println("i is :" + i);
}
```

As seen above instead of a statement we could use a block of code instead. A block of code allows us to execute multiple lines of code when a condition is filled rather than just one. We use curly brackets to illustrate the start and end of blocks.

{}

ARCTICTIGER

Semicolons

- A semicolon should be placed at the end of ever statement.
- A block is not a statement and should not have a semicolon after it.

ARCTICTIGER

Exercise 2

Rewrite your program created in exercise 1.

In this program use a for loop to write out a table in terminal showing the result of converting 1 - 50 gallons to liters.

ARCTICTIGER