Introduction to Java

ICS 2 – Introduction to Computer Programming

Brief History of Java

- Earlier called Oak Programming Language
- Created by James Gosling of Sun Microsystems
- Later called Java due to copyright reasons
- Platform independent language

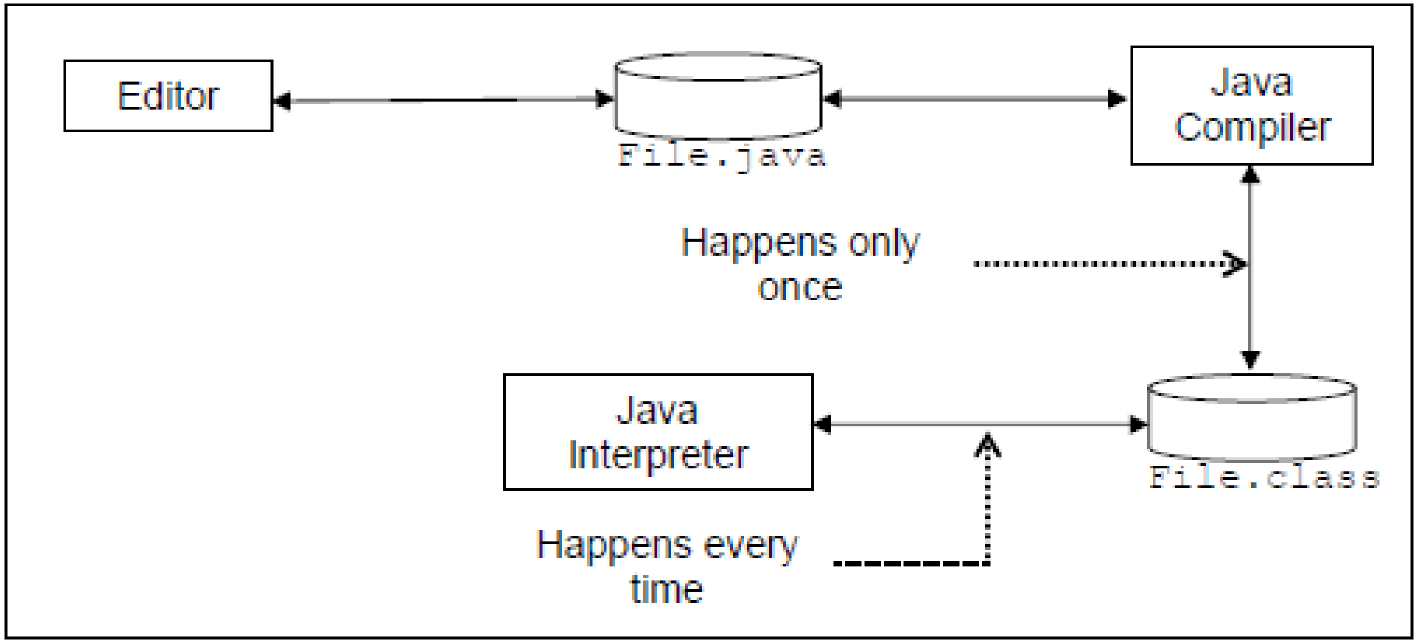
Java Technology

- Programming Language
  - Can generate all kinds of application
- Development Environment
  - Provides developer tools
- Application Environment
  - Can run on a machine where JRE is installed
- Deployment Environment
  - SDK contains complete set of class files for all types of Java technology packages

Features of Java

- Java Virtual Machine (JVM)
  - An imaginary machine used to emulate software on a real machine
- Garbage Collection
  - Responsible for deallocating memory previously allocated by a programmer
- Code Security
  - Class loader loads needed classes
  - Byte code verifier tests the format of the code
  - Code is executed

Phases of a Java Program



Compiling and Executing a Program

- Compiling
  - The process of converting a source code into a machine readable format (class)
- Executing
  - The process of running the program to see the output of that program

Types of Errors

1. Syntax Errors
   - Errors in form (syntax)
   - Inadvertently committed while typing the source code
   - Checked and reported by the compiler
2. Run-Time Error
   - Errors in meaning (semantics)
   - Sometimes referred to as Logical Errors

Java Programming Construct

- class Declaration

public class <*class_name*> {

        statements…

}

- main( ) method

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

        statements….

}
```

Comments

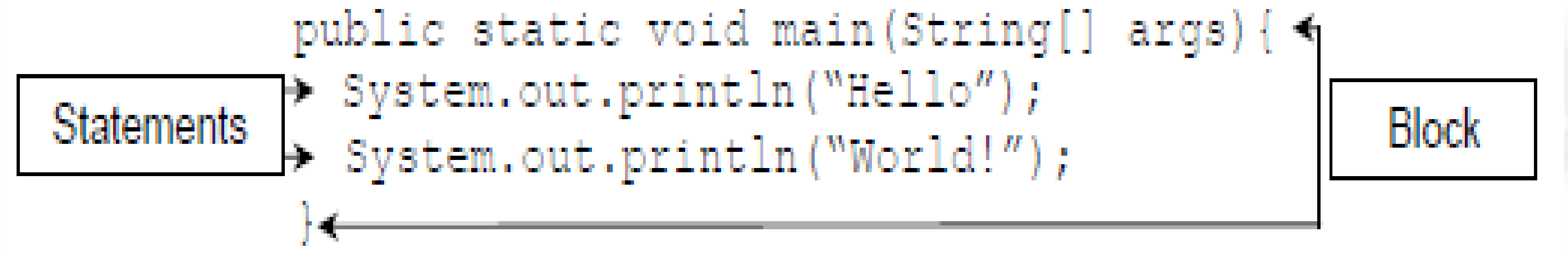
- Used to document a part of the code.
- Not part of the program itself and is only used for documentation purposes
- Types:
  - Single line comment – denoted by the symbol //
  - Multi-line comment – enclosed in symbols /* … */

Printing Elements on Screen

- Java uses *System.out.print/ln( )* method to display elements on screen.
  - System.out.println – displays elements contained on its parameter per line
  - System.out.print - – displays elements contained on its parameter on a single line

Java Statement Blocks

- Statement
  - Also called instruction is one or more lines of codes terminated by a semicolon
- Block
  - One or more statements bound by an open and close curly brace



```
public static void main(String[] args){
 System.out.println("Hello");
 System.out.println("World!");
}
```

Java Identifiers

- Also called tokens
- Usually a user defined name that represent labels of variables, methods, and classes
- Identifiers are case sensitive, it should be used the way it was written
- Cannot be any of the reserved words used in Java

Conventions in Declaring Variables

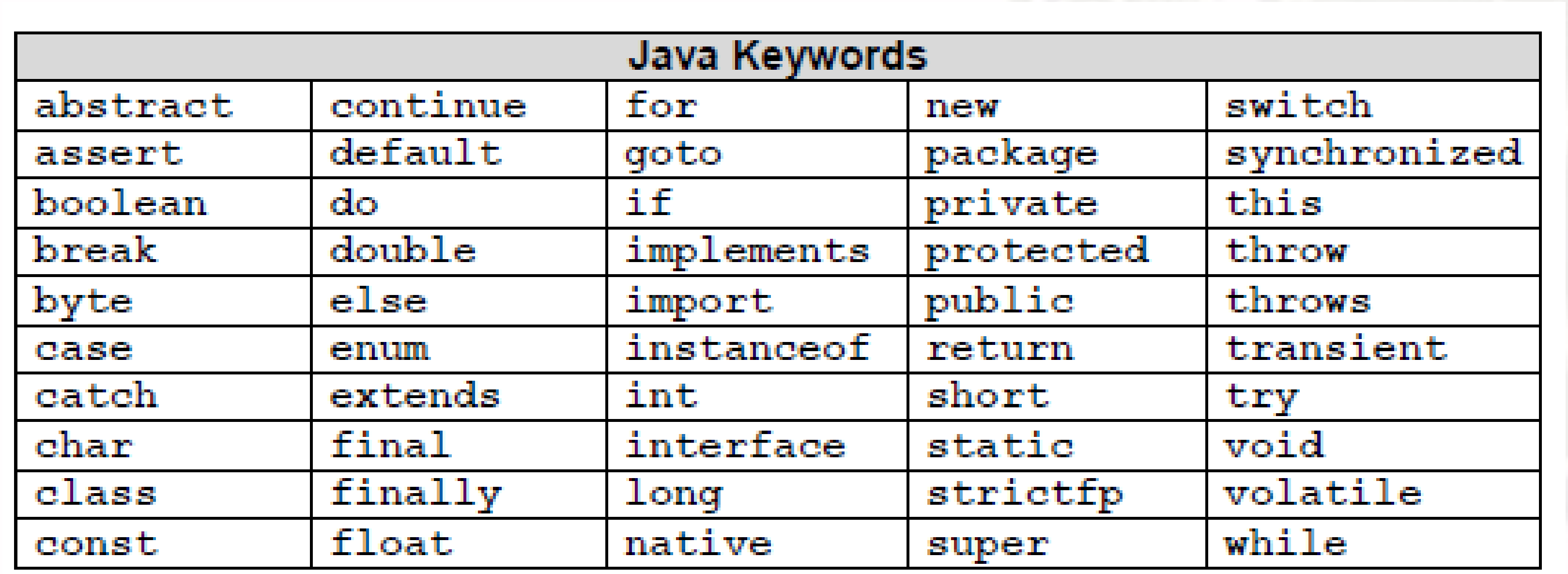
1. Names are made up of letters and numbers
2. The first character should always start with a letter
3. Java is case sensitive
4. The underscore symbol _ and dollar sign $ are considered letters in Java but not recommended to be used as a first character in a name

Java Coding Conventions

1. For names of classes, capitalize the first letter of the class name
2. For method and variable names, the first letter of the word should start with a small letter
3. For multi-word identifiers, capitalize the first letter of each word except the first word

Java Keywords

- Keywords are predefined identifiers reserved by Java for a specific purpose.
- Cannot be used as identifiers



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

Java Literals and Constants

- Integer Literals
  - Can be represented in decimal (base 10), hexadecimal (base 16) and octal (base 8) number systems
  - Examples:

    12 (decimal)
    0xC (hexadecimal)
    014 (octal)
- Floating Point Literals
  - Represent decimals with fractional part
  - Examples

    3.1416
    5.8234e2

- Boolean Literals
  - Only have two values, true (1) or false (0)
- Character Literals
  - Represents a single Unicode symbol
  - Should be enclosed in single quote delimiter
  - Example:

'a'	Letter a
'z'	Letter z
'\n'	new line character

- String Literals
  - Represents multiple characters and are enclosed by double quotes
  - Example:

    “Hello World”

    “Java Programming”

Primitive Data Types

- Determine the range a certain variable can hold and set of operations that are defined for values of this type
- It specifies:
    - The kind of values that can be assumed by a variable of that type
    - The range of values that can be assumed by a variable of that type
    - The amount of memory needed by a variable to store a value of that type

Types of Primitive Data Types

- Logical – boolean
  - Stores only two values; either “true” or “false”
  - Example:

```
boolean state = true;
Boolean value;
```

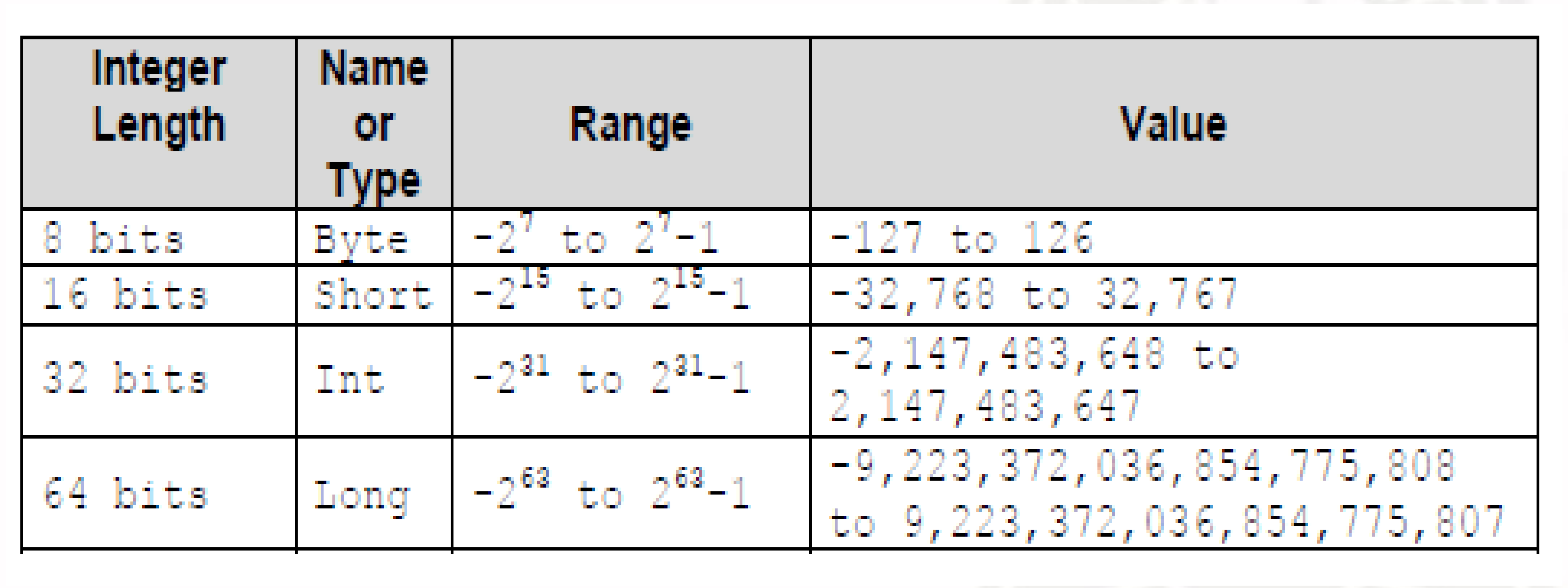
- Textual – char
  - Represents a single Unicode character
  - Example:

```
char letter = ‘A’;
char choice;
```

- Integral – byte, short, int and long
  - Uses three forms – decimal, octal or hexadecimal
  - Default type is *int,* but can also be defined as *byte*, *long* or *short*
  - Difference between the types lies on the length of values it can store
  - Example

```
byte no = 10;
int number = 500
short x;
long z;
```

Integral Data Type Range



Integer Length	Name or Type	Range	Value
8 bits	Byte	-2^7 to 2^7-1	-127 to 126
16 bits	Short	-2^{15} to $2^{15}-1$	-32,768 to 32,767
32 bits	Int	-2^{31} to $2^{31}-1$	-2,147,483,648 to 2,147,483,647
64 bits	Long	-2^{63} to $2^{63}-1$	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

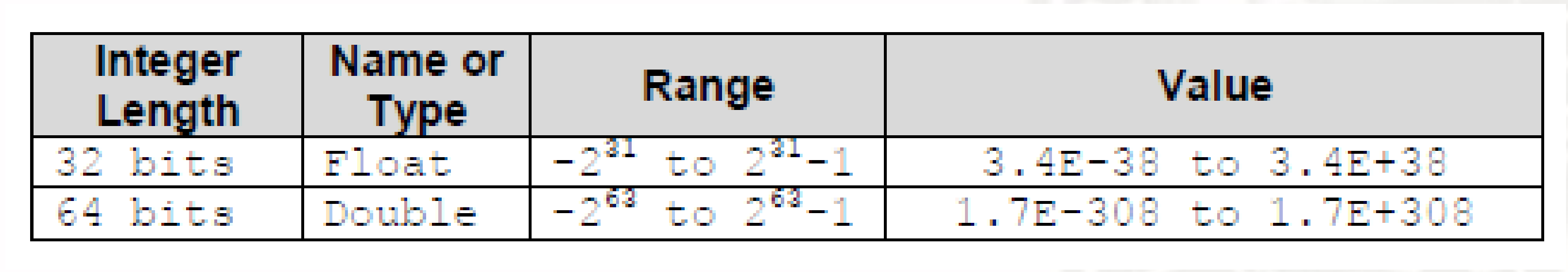
- Floating Point – float and double
  - Used as a data type that stores and processes number with decimal places or contains either one of the following:

E or e	- exponential value
F or f	- float
D or d	- double

  - Example:

```
float pi = 3.1416;
double decimal = 123456.73e10D;
```

- Floating point types and ranges



Integer Length	Name or Type	Range	Value
32 bits	Float	-2^{31} to $2^{31}-1$	3.4E-38 to 3.4E+38
64 bits	Double	-2^{63} to $2^{63}-1$	1.7E-308 to 1.7E+308

String Class

- Contains a series or multiple characters whose literal is enclosed in double quotes
- Not a *primitive data type* but a *reference type*
- Example:

```
        String subject = “CCS.1101”;
        String college = “CITCS”;
        String name;
```

Program Variables

- Entities where data can be stored to
- Values stored can change anytime
- Characteristics:
  - Name – referred to as identifier
  - Address – portion of RAM allocated
  - Value – content of the memory cell
  - Data Type – range of values it can store
  - Lifetime – time at which the variable is bound to a specific memory location
  - Scope – declaration until termination

Declaring and Initializing Variables

- Syntax:

  <data type> <name> [=<initial value>];

  - Values enclosed in <> are required while those inside[ ] are optional.