



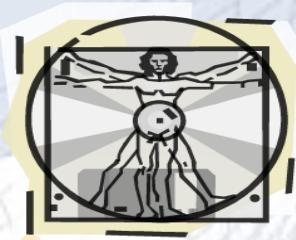
Computer Programming

Introduction

Özgür Koray SAHİNGÖZ
Prof.Dr.

Biruni University
Computer Engineering Department

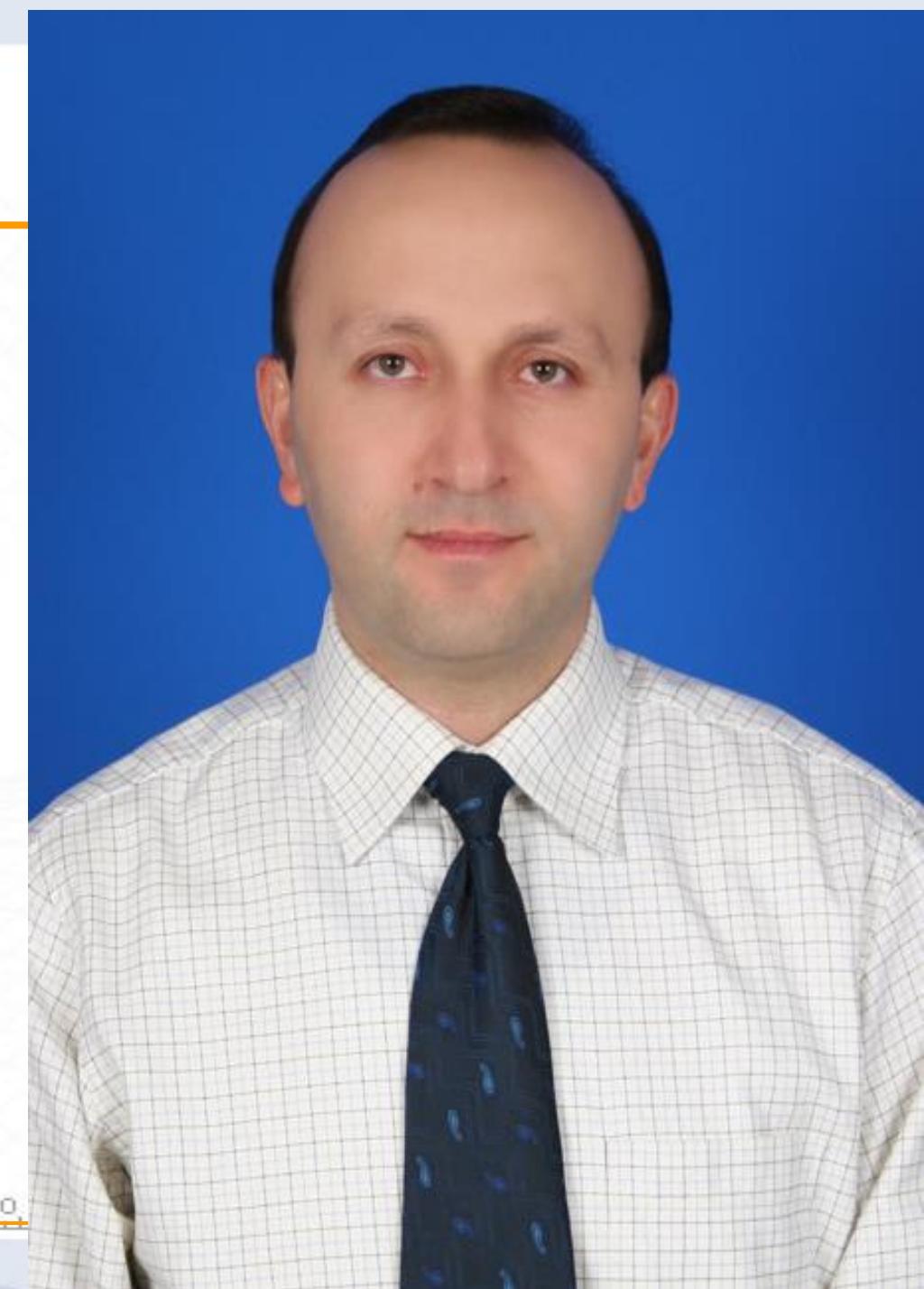


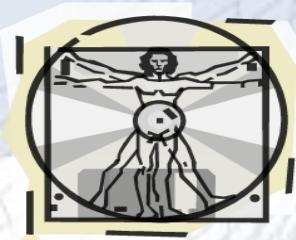


Instructor Info

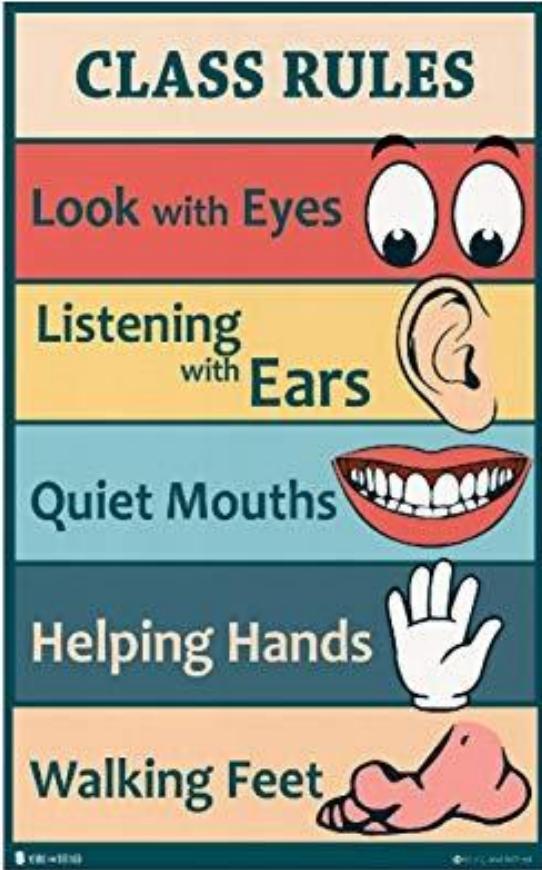
- Email: [**osahingoz@biruni.edu.tr**](mailto:osahingoz@biruni.edu.tr)
- Phone: +90 212 409 1212 -1538
- Office: R833
- Office hours: Tuesday 13:00-15:00
- Responsive with email
- Course Webpage:

Follow the BEM.Biruni.edu.tr

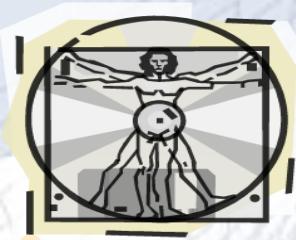




Some Important Rules



0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300
5400
5500
5600
5700
5800
5900
6000
6100
6200
6300
6400
6500
6600
6700
6800
6900
7000
7100
7200
7300
7400
7500
7600
7700
7800
7900
8000
8100
8200
8300
8400
8500
8600
8700
8800
8900
9000
9100
9200
9300
9400
9500
9600
9700
9800
9900
10000
10100
10200
10300
10400
10500
10600
10700
10800
10900
11000
11100
11200
11300
11400
11500
11600
11700
11800
11900
12000
12100
12200
12300
12400
12500
12600
12700
12800
12900
13000
13100
13200
13300
13400
13500
13600
13700
13800
13900
14000
14100
14200
14300
14400
14500
14600
14700
14800
14900
15000
15100
15200
15300
15400
15500
15600
15700
15800
15900
16000
16100
16200
16300
16400
16500
16600
16700
16800
16900
17000
17100
17200
17300
17400
17500
17600
17700
17800
17900
18000
18100
18200
18300
18400
18500
18600
18700
18800
18900
19000
19100
19200
19300
19400
19500
19600
19700
19800
19900
20000
20100
20200
20300
20400
20500
20600
20700
20800
20900
21000
21100
21200
21300
21400
21500
21600
21700
21800
21900
22000
22100
22200
22300
22400
22500
22600
22700
22800
22900
23000
23100
23200
23300
23400
23500
23600
23700
23800
23900
24000
24100
24200
24300
24400
24500
24600
24700
24800
24900
25000
25100
25200
25300
25400
25500
25600
25700
25800
25900
26000
26100
26200
26300
26400
26500
26600
26700
26800
26900
27000
27100
27200
27300
27400
27500
27600
27700
27800
27900
28000
28100
28200
28300
28400
28500
28600
28700
28800
28900
29000
29100
29200
29300
29400
29500
29600
29700
29800
29900
30000
30100
30200
30300
30400
30500
30600
30700
30800
30900
31000
31100
31200
31300
31400
31500
31600
31700
31800
31900
32000
32100
32200
32300
32400
32500
32600
32700
32800
32900
33000
33100
33200
33300
33400
33500
33600
33700
33800
33900
34000
34100
34200
34300
34400
34500
34600
34700
34800
34900
35000
35100
35200
35300
35400
35500
35600
35700
35800
35900
36000
36100
36200
36300
36400
36500
36600
36700
36800
36900
37000
37100
37200
37300
37400
37500
37600
37700
37800
37900
38000
38100
38200
38300
38400
38500
38600
38700
38800
38900
39000
39100
39200
39300
39400
39500
39600
39700
39800
39900
40000
40100
40200
40300
40400
40500
40600
40700
40800
40900
41000
41100
41200
41300
41400
41500
41600
41700
41800
41900
42000
42100
42200
42300
42400
42500
42600
42700
42800
42900
43000
43100
43200
43300
43400
43500
43600
43700
43800
43900
44000
44100
44200
44300
44400
44500
44600
44700
44800
44900
45000
45100
45200
45300
45400
45500
45600
45700
45800
45900
46000
46100
46200
46300
46400
46500
46600
46700
46800
46900
47000
47100
47200
47300
47400
47500
47600
47700
47800
47900
48000
48100
48200
48300
48400
48500
48600
48700
48800
48900
49000
49100
49200
49300
49400
49500
49600
49700
49800
49900
50000
50100
50200
50300
50400
50500
50600
50700
50800
50900
51000
51100
51200
51300
51400
51500
51600
51700
51800
51900
52000
52100
52200
52300
52400
52500
52600
52700
52800
52900
53000
53100
53200
53300
53400
53500
53600
53700
53800
53900
54000
54100
54200
54300
54400
54500
54600
54700
54800
54900
55000
55100
55200
55300
55400
55500
55600
55700
55800
55900
56000
56100
56200
56300
56400
56500
56600
56700
56800
56900
57000
57100
57200
57300
57400
57500
57600
57700
57800
57900
58000
58100
58200
58300
58400
58500
58600
58700
58800
58900
59000
59100
59200
59300
59400
59500
59600
59700
59800
59900
60000
60100
60200
60300
60400
60500
60600
60700
60800
60900
61000
61100
61200
61300
61400
61500
61600
61700
61800
61900
62000
62100
62200
62300
62400
62500
62600
62700
62800
62900
63000
63100
63200
63300
63400
63500
63600
63700
63800
63900
64000
64100
64200
64300
64400
64500
64600
64700
64800
64900
65000
65100
65200
65300
65400
65500
65600
65700
65800
65900
66000
66100
66200
66300
66400
66500
66600
66700
66800
66900
67000
67100
67200
67300
67400
67500
67600
67700
67800
67900
68000
68100
68200
68300
68400
68500
68600
68700
68800
68900
69000
69100
69200
69300
69400
69500
69600
69700
69800
69900
70000
70100
70200
70300
70400
70500
70600
70700
70800
70900
71000
71100
71200
71300
71400
71500
71600
71700
71800
71900
72000
72100
72200
72300
72400
72500
72600
72700
72800
72900
73000
73100
73200
73300
73400
73500
73600
73700
73800
73900
74000
74100
74200
74300
74400
74500
74600
74700
74800
74900
75000
75100
75200
75300
75400
75500
75600
75700
75800
75900
76000
76100
76200
76300
76400
76500
76600
76700
76800
76900
77000
77100
77200
77300
77400
77500
77600
77700
77800
77900
78000
78100
78200
78300
78400
78500
78600
78700
78800
78900
79000
79100
79200
79300
79400
79500
79600
79700
79800
79900
80000
80100
80200
80300
80400
80500
80600
80700
80800
80900
81000
81100
81200
81300
81400
81500
81600
81700
81800
81900
82000
82100
82200
82300
82400
82500
82600
82700
82800
82900
83000
83100
83200
83300
83400
83500
83600
83700
83800
83900
84000
84100
84200
84300
84400
84500
84600
84700
84800
84900
85000
85100
85200
85300
85400
85500
85600
85700
85800
85900
86000
86100
86200
86300
86400
86500
86600
86700
86800
86900
87000
87100
87200
87300
87400
87500
87600
87700
87800
87900
88000
88100
88200
88300
88400
88500
88600
88700
88800
88900
89000
89100
89200
89300
89400
89500
89600
89700
89800
89900
90000
90100
90200
90300
90400
90500
90600
90700
90800
90900
91000
91100
91200
91300
91400
91500
91600
91700
91800
91900
92000
92100
92200
92300
92400
92500
92600
92700
92800
92900
93000
93100
93200
93300
93400
93500
93600
93700
93800
93900
94000
94100
94200
94300
94400
94500
94600
94700
94800
94900
95000
95100
95200
95300
95400
95500
95600
95700
95800
95900
96000
96100
96200
96300
96400
96500
96600
96700
96800
96900
97000
97100
97200
97300
97400
97500
97600
97700
97800
97900
98000
98100
98200
98300
98400
98500
98600
98700
98800
98900
99000
99100
99200
99300
99400
99500
99600
99700
99800
99900
100000

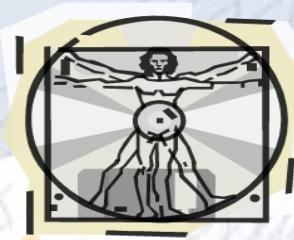


Reminders

- Please turn off and remove your cell phones.



© Can Stock Photo - csp8472702

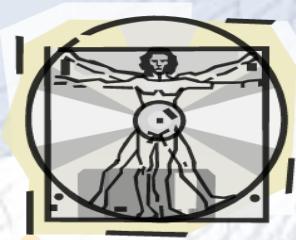


Use of Cell Phones in the Lecture is Strictly Forbidden

- My mother will call me!!!
 - I'm waiting very important call
 - I need to look at the meaning of some words
 - I use it as watch

-1 point for each attempt





Use of Tablet or Laptop is allowed (only for looking at the Lecture Notes)

**No Typing
No Keyboard Use**





Late Entrance to Lecture is Allowed



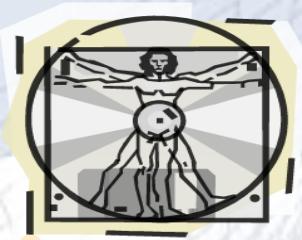
0
100
200
300
400
500

|0|50|100|150|200|250|300|350|400|450|



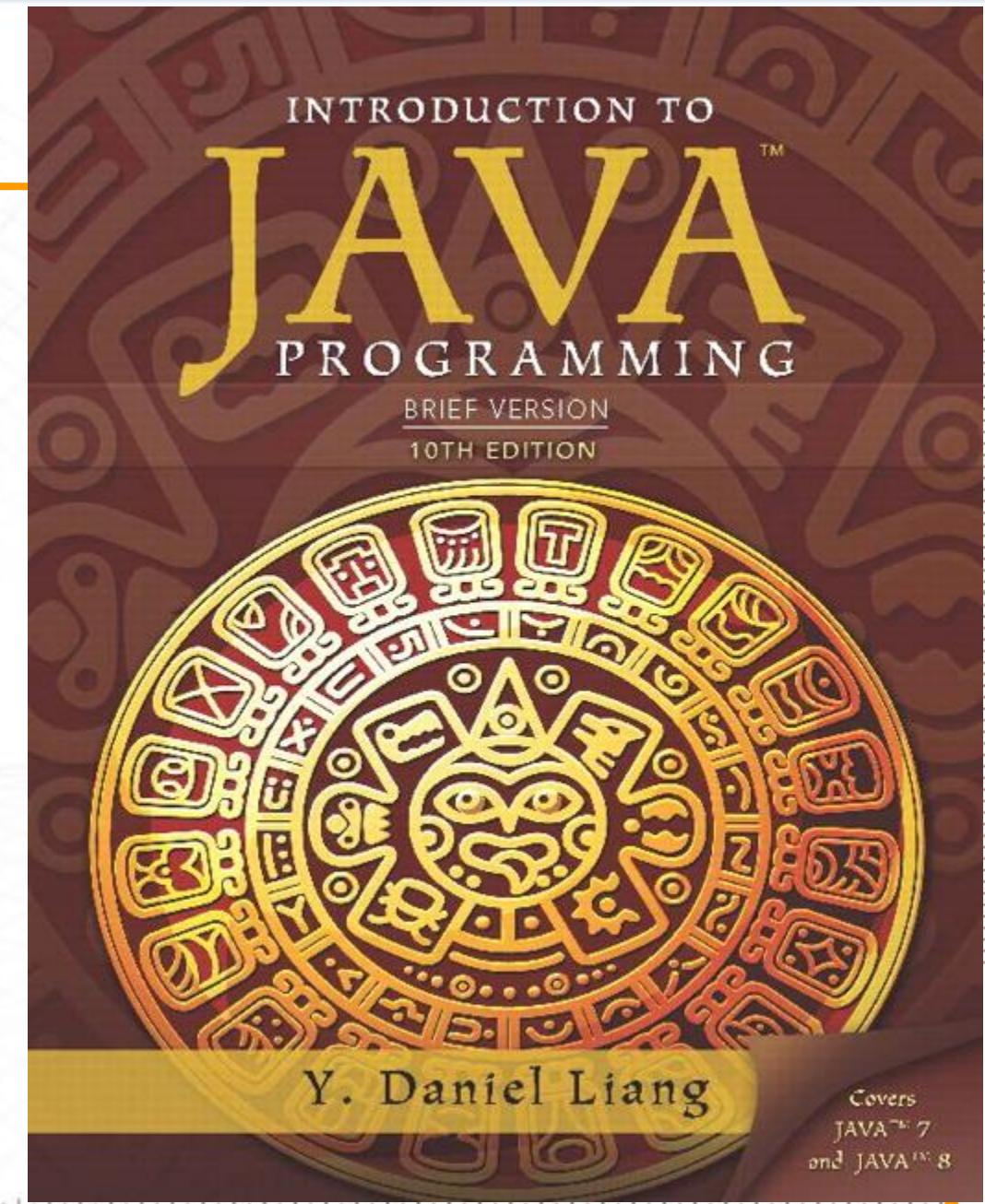
If you leave the Class you cannot reenter in the
SAME HOUR

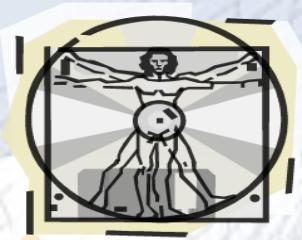




Textbook

- **ASIN** : 0134670949
- **Publisher** : Pearson; 11. basım (1 Mart 2017)
- **Language** : İngilizce
- **ISBN-10** : 9780134670942
- **ISBN-13** : 978-0134670942
- **Dimension** : 4.06 x 20.07 x 25.65 cm
- <https://www.amazon.com.tr/Introduction-Programming-Structures-Comprehensive-Version/dp/0134670949>

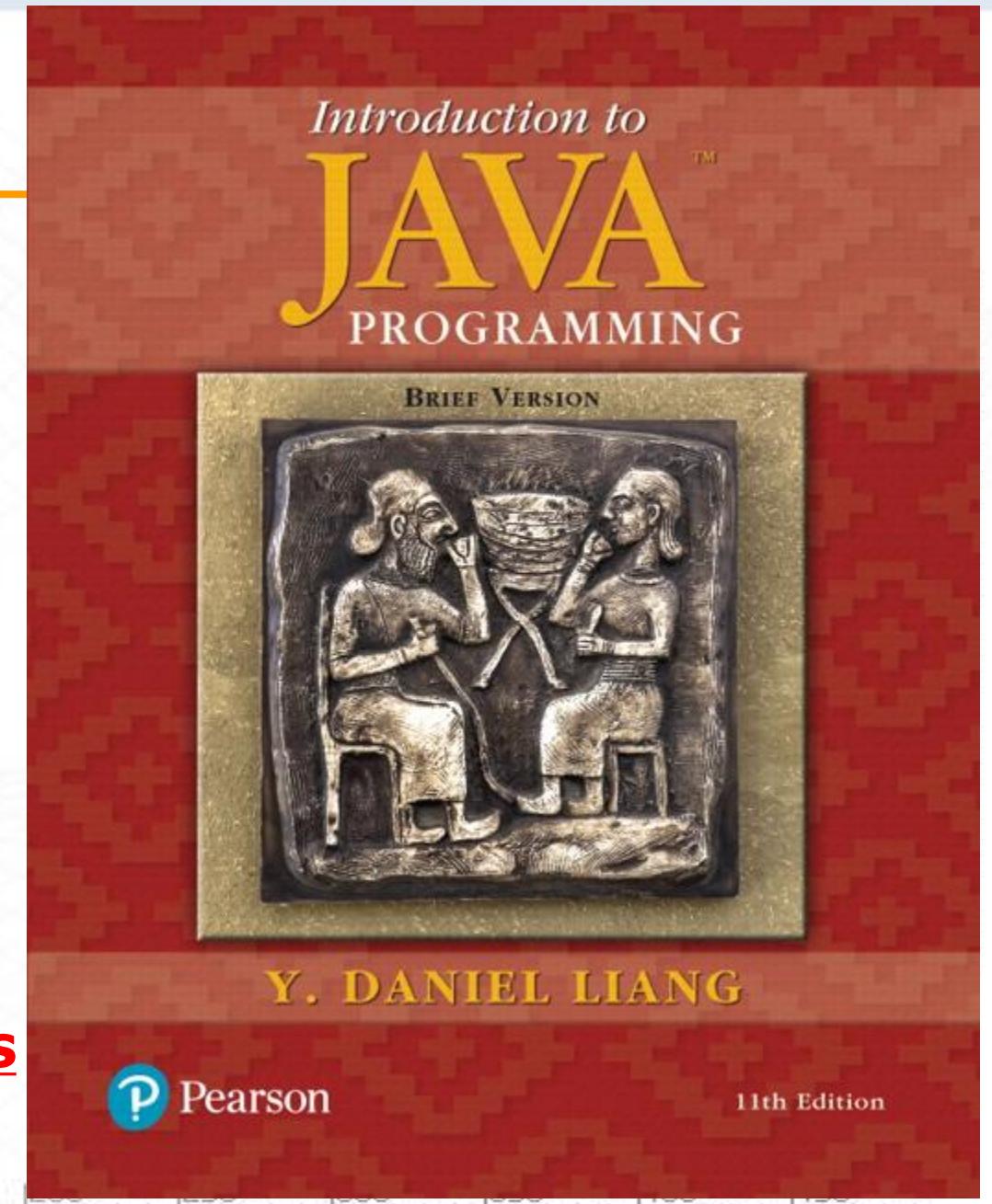


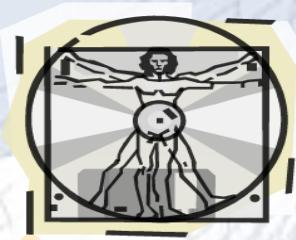


Textbook

- Introduction to Java Programming,
- Brief Version, 11/E
- Liang
- ISBN-10: 0134611039
- ISBN-13: 9780134611037

<https://media.pearsoncmg.com/bc/abp/cs-resources/products/series.html#series,series=Liang>

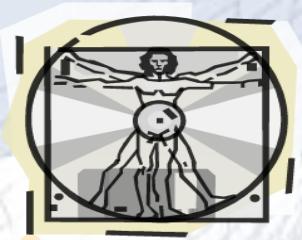




Content

- Introduction to Computers, Programs and Java
- Elementary Programming
- Selections
- Mathematical Functions, Characters, and Strings
- **MIDTERM**
- Loops
- Methods
- Single-Dimensional Arrays
- Multidimensional Arrays
- **FINAL**





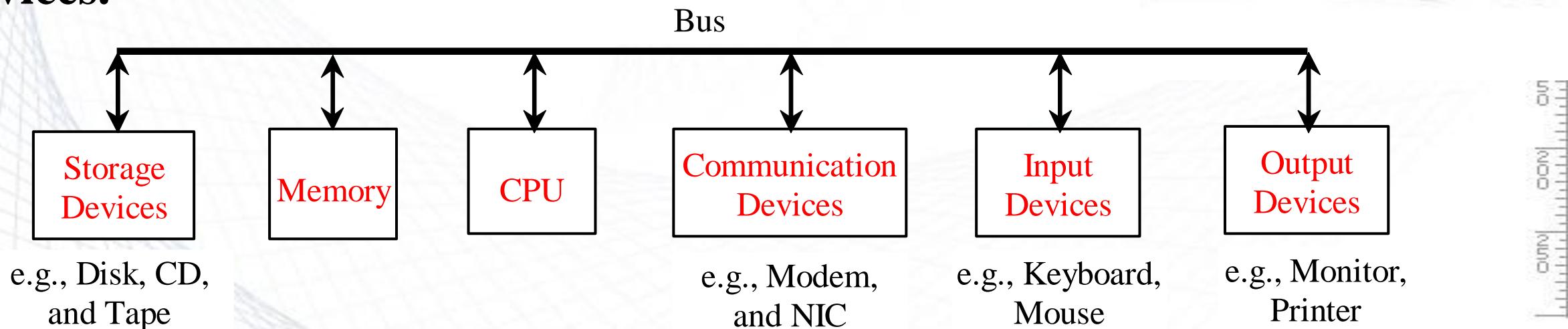
Grades

■ Midterm	30
■ Projects/Homeworks/Quiz	20
■ Final	50
	+
Total	100

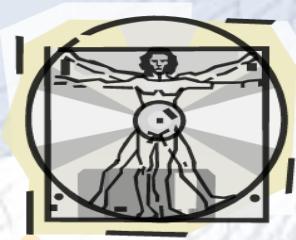


What is a Computer?

A computer consists of a CPU, memory, hard disk, floppy disk, monitor, printer, and communication devices.

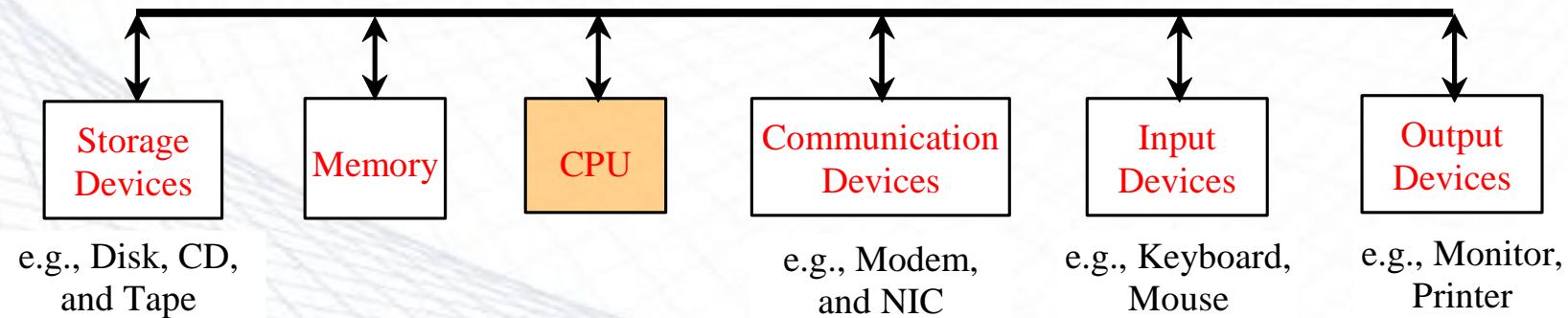


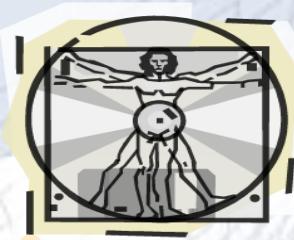
A computer consists of a CPU, memory, hard disk, floppy disk, monitor, printer, and communication devices.



CPU

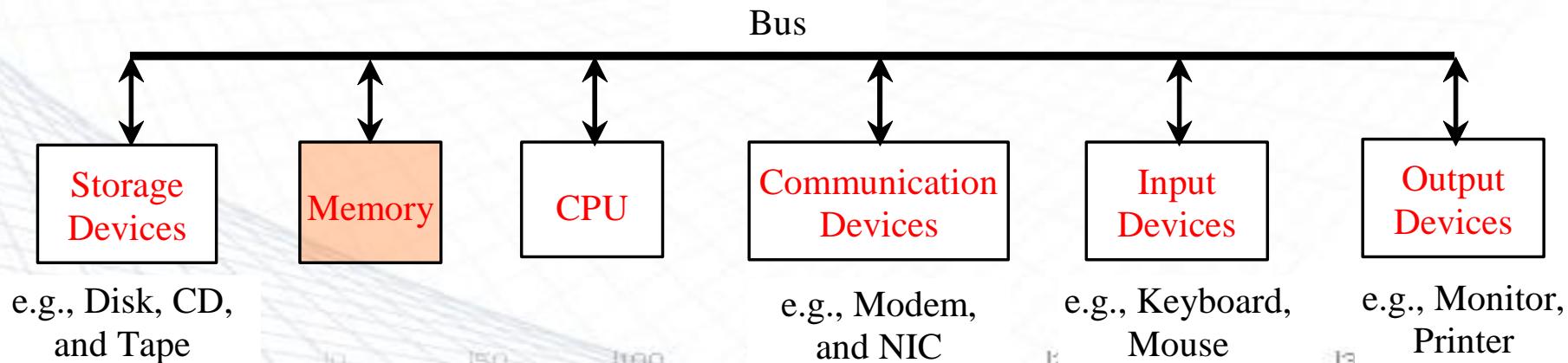
The central processing unit (CPU) is the brain of a computer. It retrieves instructions from memory and executes them. The CPU speed is measured in megahertz (MHz), with 1 megahertz equaling 1 million pulses per second. The speed of the CPU has been improved continuously. If you buy a PC now, you can get an Intel Pentium 4 Processor at 3 gigahertz (1 gigahertz is 1000 megahertz).

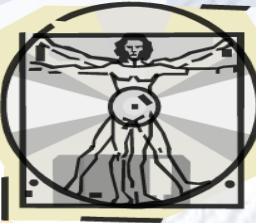




Memory

Memory is to store data and program instructions for CPU to execute. A memory unit is an ordered sequence of bytes, each holds eight bits. A program and its data must be brought to memory before they can be executed. A memory byte is never empty, but its initial content may be meaningless to your program. The current content of a memory byte is lost whenever new information is placed in it.

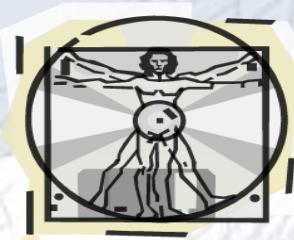




How Data is Stored?

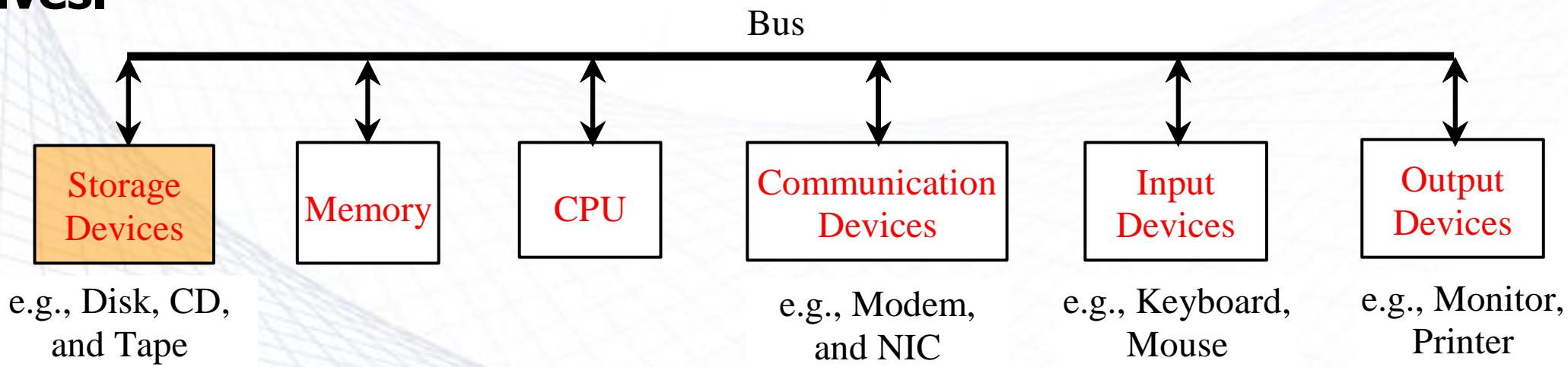
Data of various kinds, such as numbers, characters, and strings, are encoded as a series of bits (zeros and ones). Computers use zeros and ones because digital devices have two stable states, which are referred to as *zero* and *one* by convention. The programmers need not to be concerned about the encoding and decoding of data, which is performed automatically by the system based on the encoding scheme. The encoding scheme varies. For example, character ‘J’ is represented by 01001010 in one byte. A small number such as three can be stored in a single byte. If computer needs to store a large number that cannot fit into a single byte, it uses a number of adjacent bytes. No two data can share or split a same byte. A byte is the minimum storage unit.

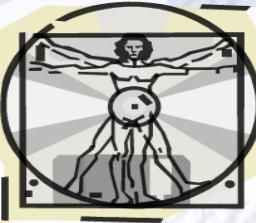
	Memory address	Memory content	
	↓	↓	
	.	.	
	.	.	
	.	.	
2000	01001010	Encoding for character ‘J’	
2001	01100001	Encoding for character ‘a’	
2002	01110110	Encoding for character ‘v’	
2003	01100001	Encoding for character ‘a’	
2004	00000011	Encoding for number 3	



Storage Devices

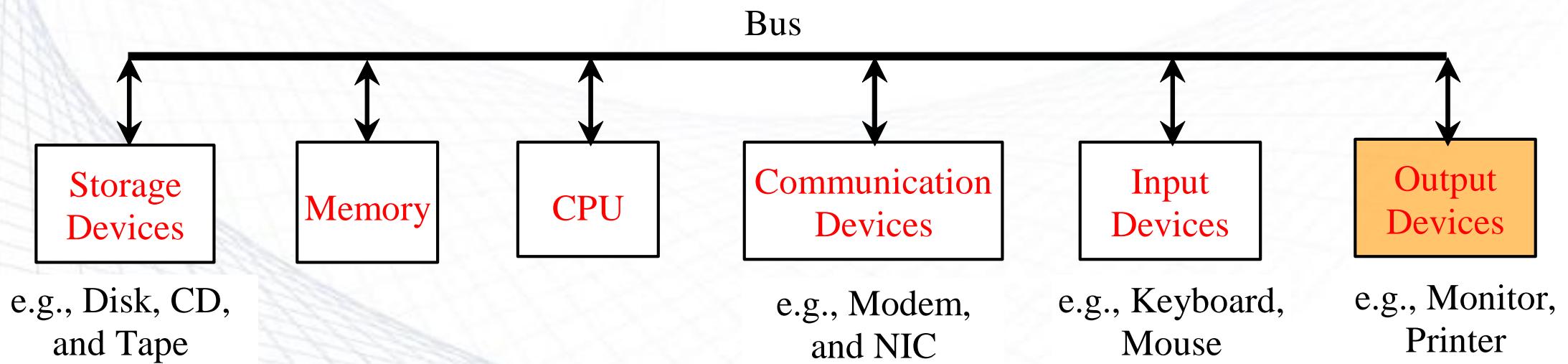
Memory is volatile, because information is lost when the power is off. Programs and data are permanently stored on storage devices and are moved to memory when the computer actually uses them. There are three main types of storage devices: Disk drives (hard disks and floppy disks), CD drives (CD-R and CD-RW), and Tape drives.

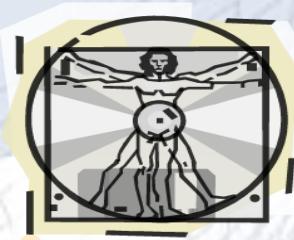




Output Devices: Monitor

The monitor displays information (text and graphics). The resolution and dot pitch determine the quality of the display.





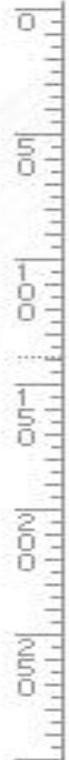
Monitor Resolution and Dot Pitch

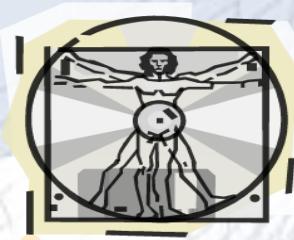
resolution

The *screen resolution* specifies the number of pixels in horizontal and vertical dimensions of the display device. *Pixels* (short for “picture elements”) are tiny dots that form an image on the screen. A common resolution for a 17-inch screen, for example, is 1,024 pixels wide and 768 pixels high. The resolution can be set manually. The higher the resolution, the sharper and clearer the image is.

dot pitch

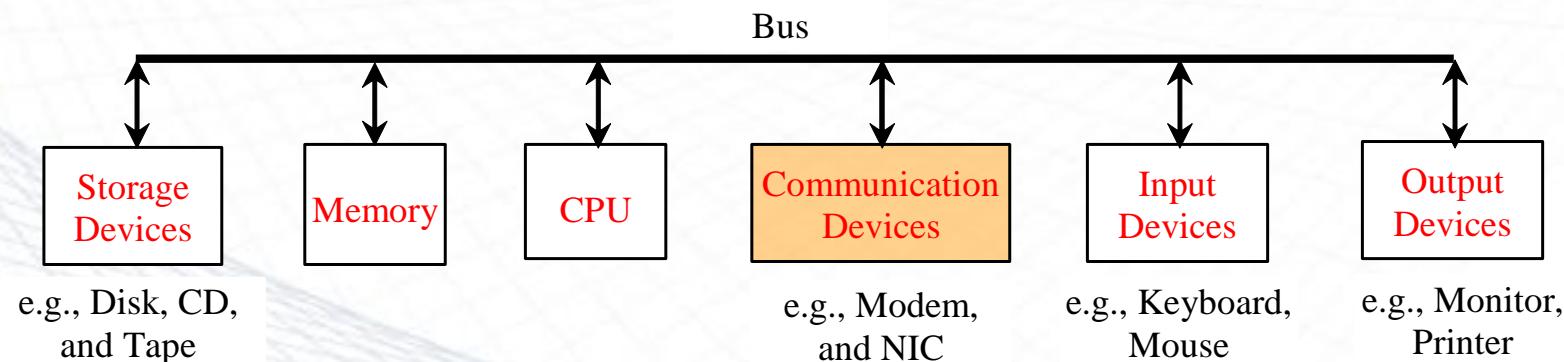
The *dot pitch* is the amount of space between pixels, measured in millimeters. The smaller the dot pitch, the sharper the display.

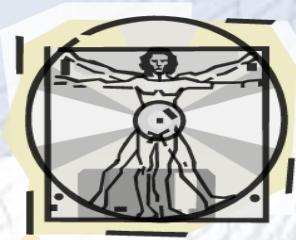




Communication Devices

A **regular modem** uses a phone line and can transfer data in a speed up to 56,000 bps (bits per second). A **DSL** (digital subscriber line) also uses a phone line and can transfer data in a speed 20 times faster than a regular modem. A **cable modem** uses the TV cable line maintained by the cable company. A cable modem is as fast as a DSL. Network interface card (**NIC**) is a device to connect a computer to a local area network (LAN). The LAN is commonly used in business, universities, and government organizations. A typical type of NIC, called **10BaseT**, can transfer data at 10 mbps (million bits per second).



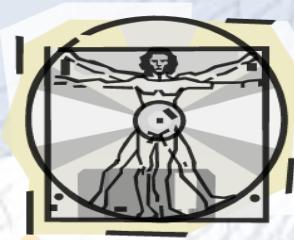


Programs

Computer *programs*, known as *software*, are instructions to the computer.

You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

Programs are written using programming languages.



Programming Languages

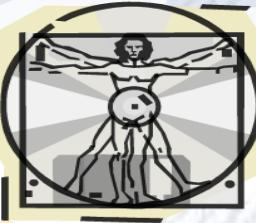
Machine Language

Assembly Language

High-Level Language

Machine language is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Program with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify. For example, to add two numbers, you might write an instruction in binary like this:

1101101010011010



Programming Languages

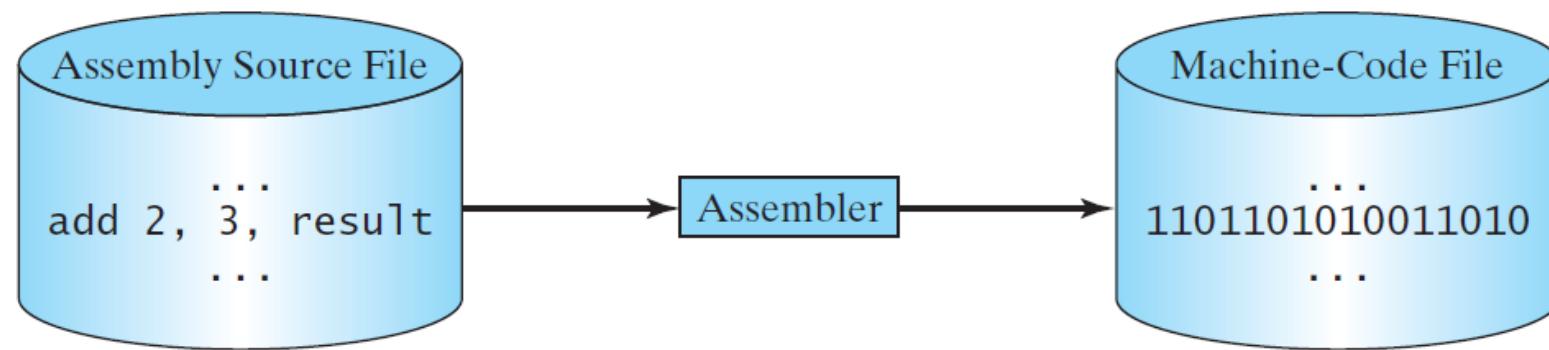
Machine Language

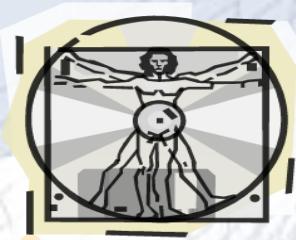
Assembly Language

High-Level Language

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code. For example, to add two numbers, you might write an instruction in assembly code like this:

ADDF3 R1, R2, R3





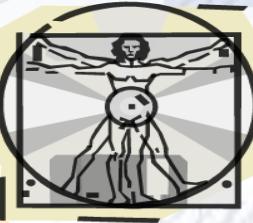
Programming Languages

Machine Language Assembly Language **High-Level Language**

The high-level languages are English-like and easy to learn and program. For example, the following is a high-level language statement that computes the area of a circle with radius 5:

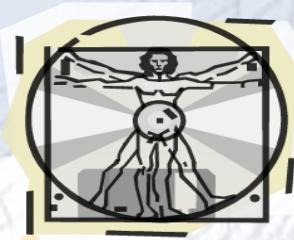
```
area = 5 * 5 * 3.1415;
```





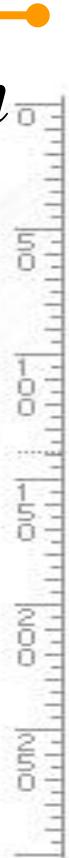
Popular High-Level Languages

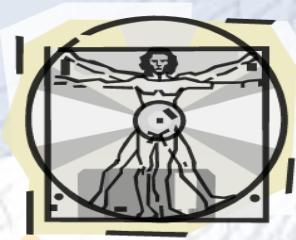
Language	Description
Ada	Named for Ada Lovelace, who worked on mechanical general-purpose computers. The Ada language was developed for the Department of Defense and is used mainly in defense projects.
BASIC	Beginner's All-purpose Symbolic Instruction Code. It was designed to be learned and used easily by beginners.
C	Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.
C++	C++ is an object-oriented language, based on C.
C#	Pronounced "C Sharp." It is a hybrid of Java and C++ and was developed by Microsoft.
COBOL	COmmon Business Oriented Language. Used for business applications.
FORTRAN	FORmula TRANslator. Popular for scientific and mathematical applications.
Java	Developed by Sun Microsystems, now part of Oracle. It is widely used for developing platform-independent Internet applications.
Pascal	Named for Blaise Pascal, who pioneered calculating machines in the seventeenth century. It is a simple, structured, general-purpose language primarily for teaching programming.
Python	A simple general-purpose scripting language good for writing short programs.
Visual Basic	Visual Basic was developed by Microsoft and it enables the programmers to rapidly develop graphical user interfaces.



Interpreting/Compiling Source Code

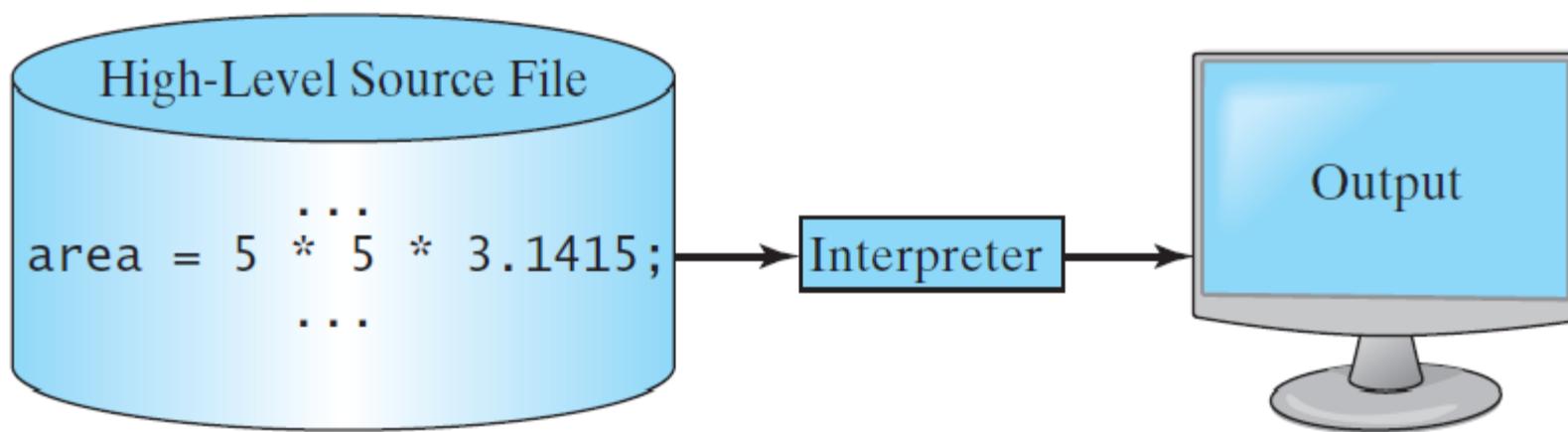
A program written in a high-level language is called a *source program* or *source code*. Because a computer cannot understand a source program, a source program must be translated into machine code for execution. The translation can be done using another programming tool called an *interpreter* or a *compiler*.

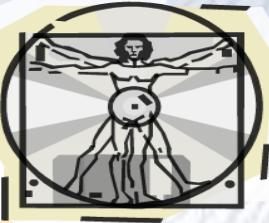




Interpreting Source Code

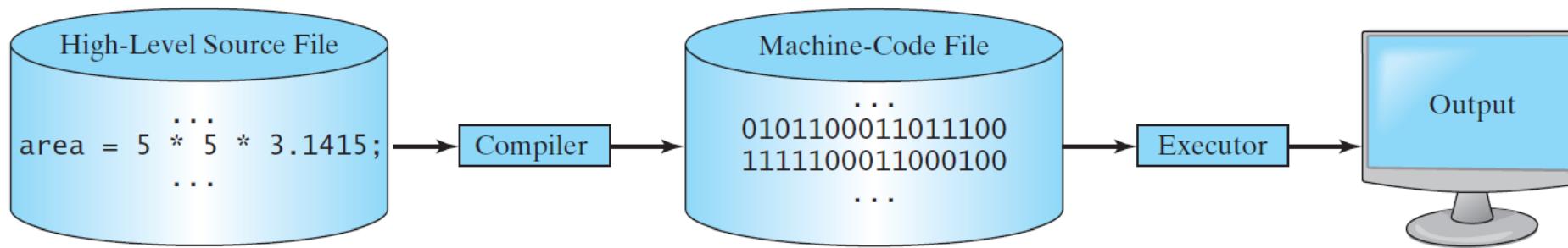
An interpreter reads one statement from the source code, translates it to the machine code or virtual machine code, and then executes it right away, as shown in the following figure. Note that a statement from the source code may be translated into several machine instructions.

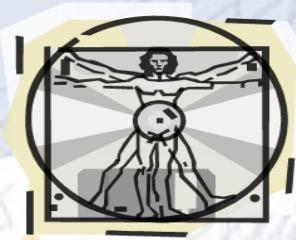




Compiling Source Code

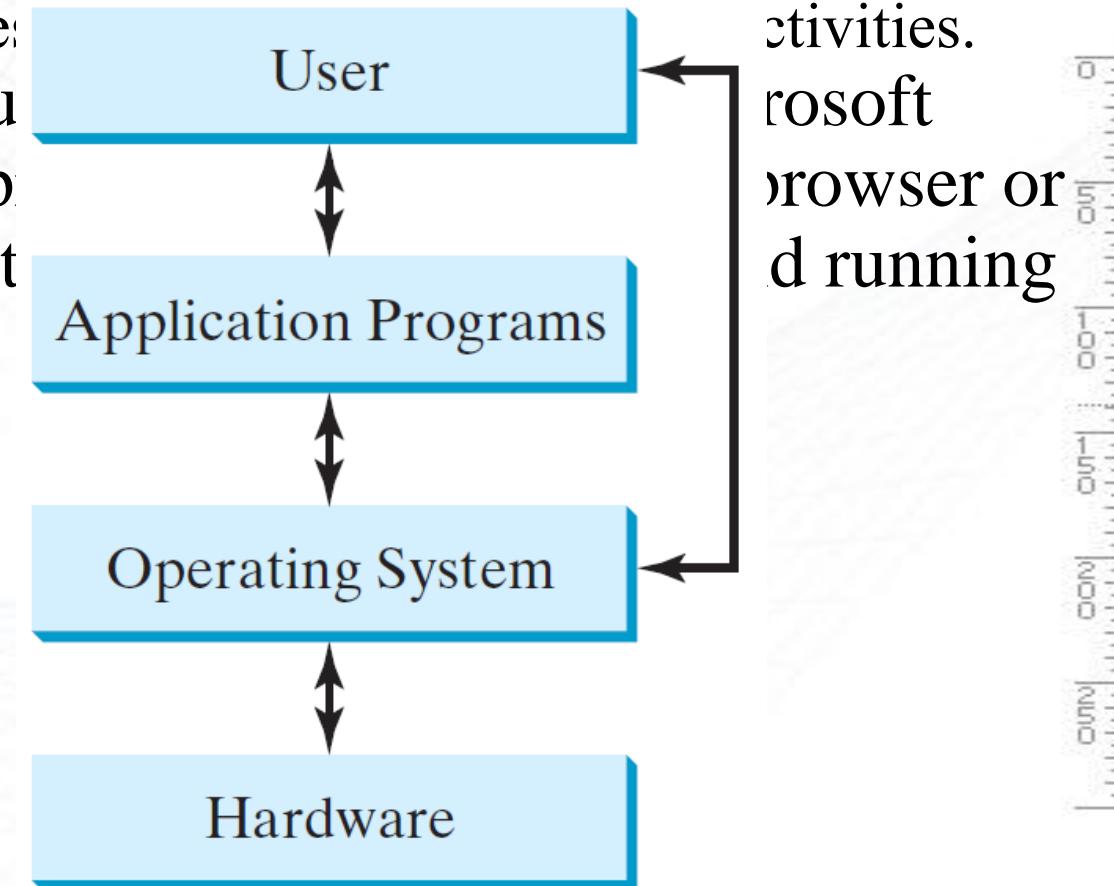
A compiler translates the entire source code into a machine-code file, and the machine-code file is then executed, as shown in the following figure.

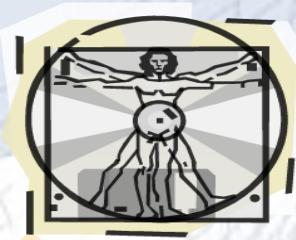




Operating Systems

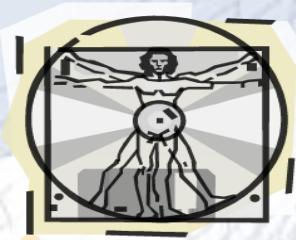
The *operating system* (OS) is a program that manages computer hardware and software resources. The popular operating systems for general-purpose computers are Microsoft Windows, Mac OS, and Linux. Application programs, such as a word processor, cannot run unless an operating system is running on the computer.





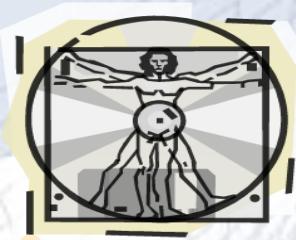
Why Java?

- The answer is that Java enables users to develop and deploy applications on the Internet for servers, desktop computers, and small hand-held devices.
- The future of computing is being profoundly influenced by the Internet, and Java promises to remain a big part of that future. Java is the Internet programming language.



Why Java

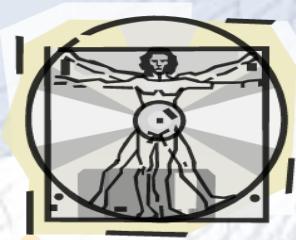
- Java is a general purpose programming language.
- Java is user-friendly
- Java for everything!
- Java is the Internet programming language.
- Java boasts of rich API
- A robust community backs Java
- Java has excellent documentation
- Java has a suite of powerful development tools



Java, Web, and Beyond

- Java can be used to develop standalone applications.
- Java can be used to develop applications running from a browser.
- Java can also be used to develop applications for hand-held devices.
- Java can be used to develop applications for Web servers.





Java's History

- James Gosling and Sun Microsystems
- Oak
- Java, May 20, 1995, Sun World
- HotJava
 - The first Java-enabled Web browser
- Early History Website:

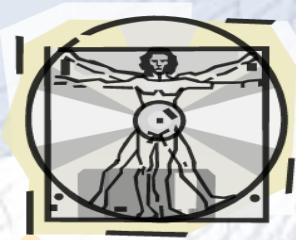
<http://www.java.com/en/javahistory/index.jsp>



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

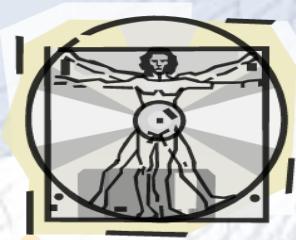
www.cs.armstrong.edu/liang/JavaCharacteristics.pdf



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is partially modeled on C++, but greatly simplified and improved. Some people refer to Java as "C++--" because it is like C++ but with more functionality and fewer negative aspects.

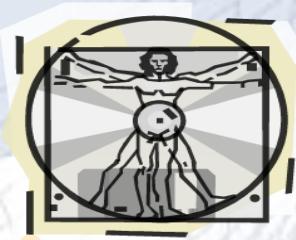


Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is inherently object-oriented. Although many object-oriented languages began strictly as procedural languages, Java was designed from the start to be object-oriented. **Object-oriented programming (OOP)** is a popular programming approach that is replacing traditional procedural programming techniques.

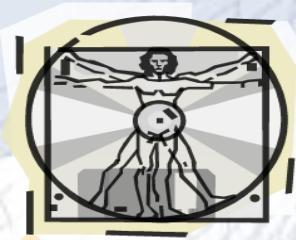
One of the central issues in software development is how to reuse code. Object-oriented programming provides great flexibility, modularity, clarity, and reusability through encapsulation, inheritance, and polymorphism.



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

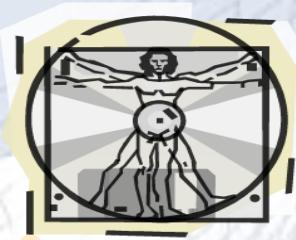
Distributed computing involves several computers working together on a network. Java is designed to make distributed computing easy. Since networking capability is inherently integrated into Java, writing network programs is like sending and receiving data to and from a file.



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

You need an interpreter to run Java programs. The programs are compiled into the Java Virtual Machine code called bytecode. The bytecode is machine-independent and can run on any machine that has a Java interpreter, which is part of the Java Virtual Machine (JVM).



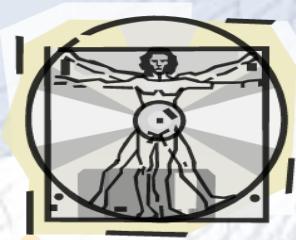
Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java compilers can detect many problems that would first show up at execution time in other languages.

Java has eliminated certain types of error-prone programming constructs found in other languages.

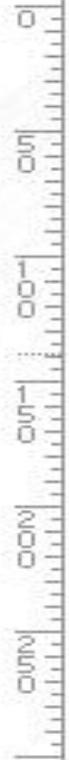
Java has a runtime exception-handling feature to provide programming support for robustness.

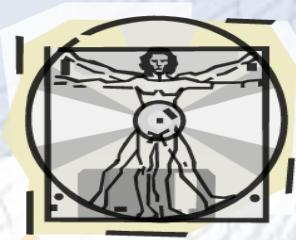


Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- **Java Is Secure**
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java implements several security mechanisms to protect your system against harm caused by stray programs.



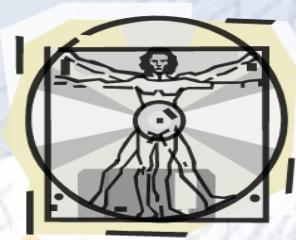


Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Write once, run anywhere

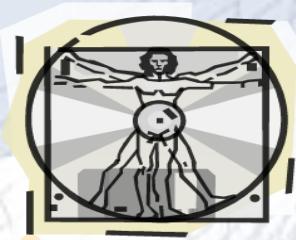
With a Java Virtual Machine (JVM), you can write one program that will run on any platform.



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

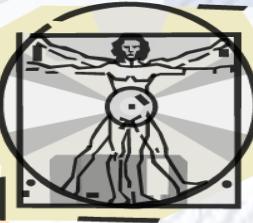
Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

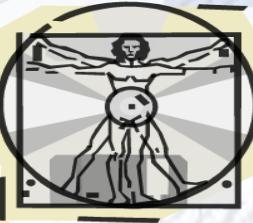
Java's performance Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Multithread programming is smoothly integrated in Java, whereas in other languages you have to call procedures specific to the operating system to enable multithreading.



Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

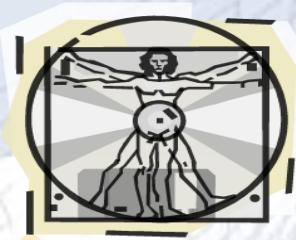
Java was designed to adapt to an evolving environment. New code can be loaded on the fly without recompilation. There is no need for developers to create, and for users to install, major new software versions. New features can be incorporated transparently as needed.



JDK Versions

- JDK 1.02 (1995)
- JDK 1.1 (1996)
- JDK 1.2 (1998)
- JDK 1.3 (2000)
- JDK 1.4 (2002)
- JDK 1.5 (2004)
- JDK 1.6 (2006)
- JDK 1.7 (2011)
- JDK 1.8 (2014)

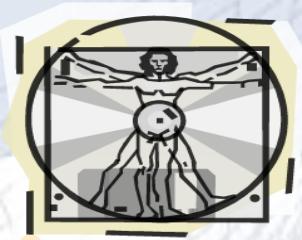
Java SE 9	September 2017	March 2018 for OpenJDK
Java SE 10	March 2018	September 2018 for OpenJDK
Java SE 11 (LTS)	September 2018	September 2026 for Azul September 2026 for IBM Semeru At least October 2024 for Eclipse Adoptium At least September 2027 for Amazon Corretto At least October 2024 for Microsoft ^{[10][11]}
Java SE 12	March 2019	September 2019 for OpenJDK
Java SE 13	September 2019	March 2020 for OpenJDK
Java SE 14	March 2020	September 2020 for OpenJDK
Java SE 15	September 2020	March 2021 for OpenJDK March 2023 for Azul ^[8]
Java SE 16	March 2021	September 2021 for OpenJDK
Java SE 17 (LTS)	September 2021	September 2029 for Azul At least September 2027 for Microsoft At least TBA for Eclipse Adoptium
Java SE 18	March 2022	September 2022 for OpenJDK
Java SE 19	September 2022	March 2023 for OpenJDK
Java SE 20	March 2023	September 2023 for OpenJDK
Java SE 21 (LTS)	September 2023	TBA



JDK Editions

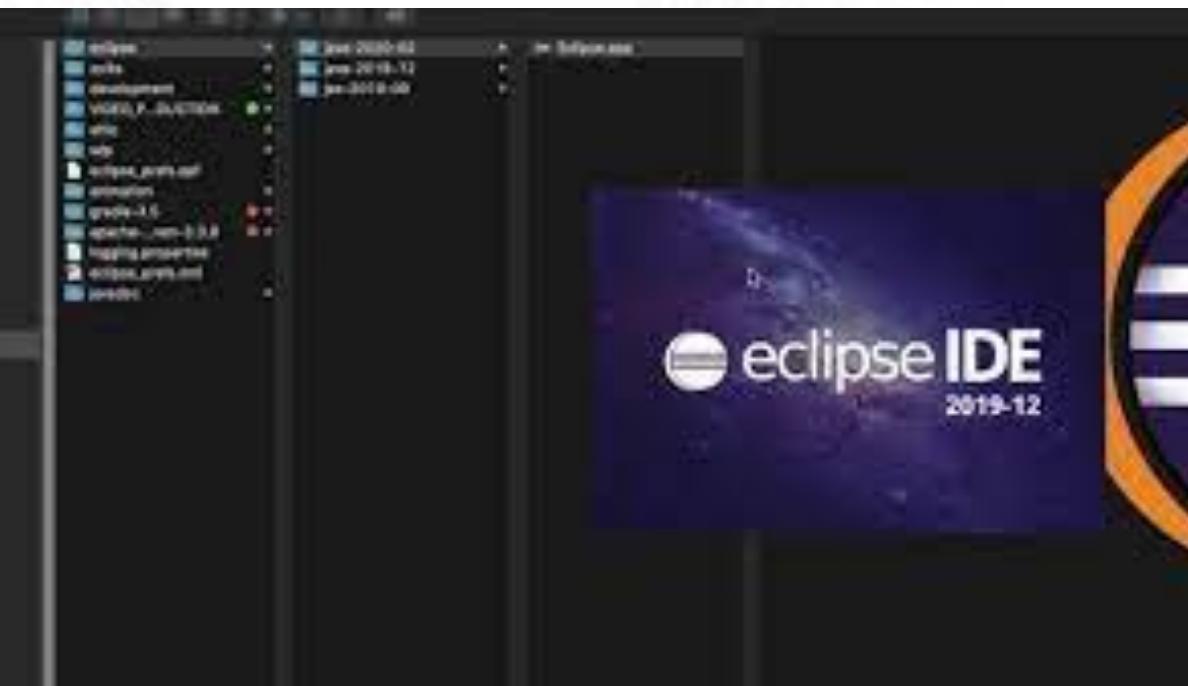
- Java Standard Edition (J2SE)
 - J2SE can be used to develop client-side standalone applications or applets.
- Java Enterprise Edition (J2EE)
 - J2EE can be used to develop server-side applications such as Java servlets, Java ServerPages, and Java ServerFaces.
- Java Micro Edition (J2ME).
 - J2ME can be used to develop applications for mobile devices such as cell phones.

This book uses J2SE to introduce Java programming.

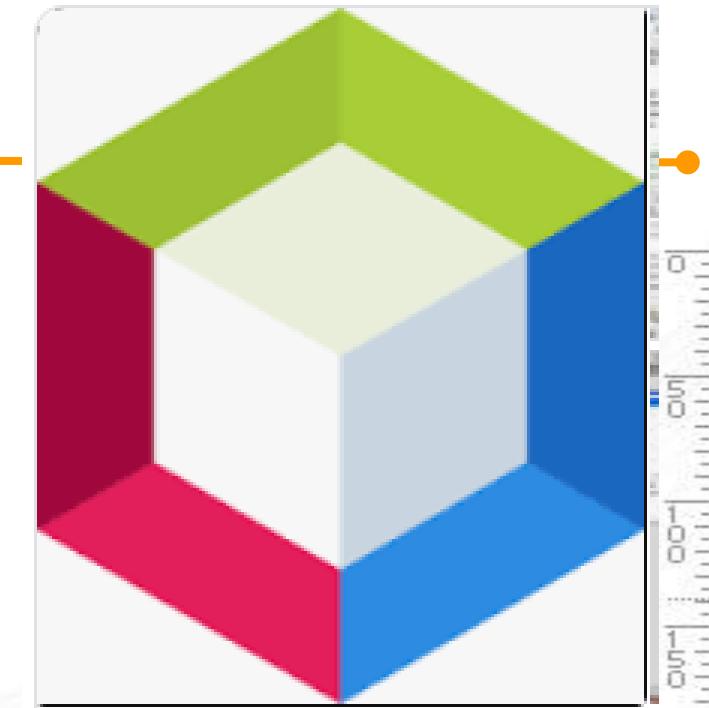


Popular Java IDEs

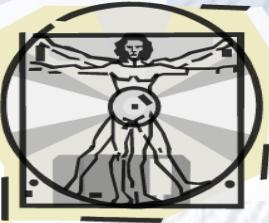
- NetBeans



- Eclipse



NetBeans
Software



A Simple Java Program

Listing 1.1

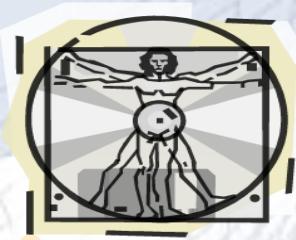
```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

Welcome

Run

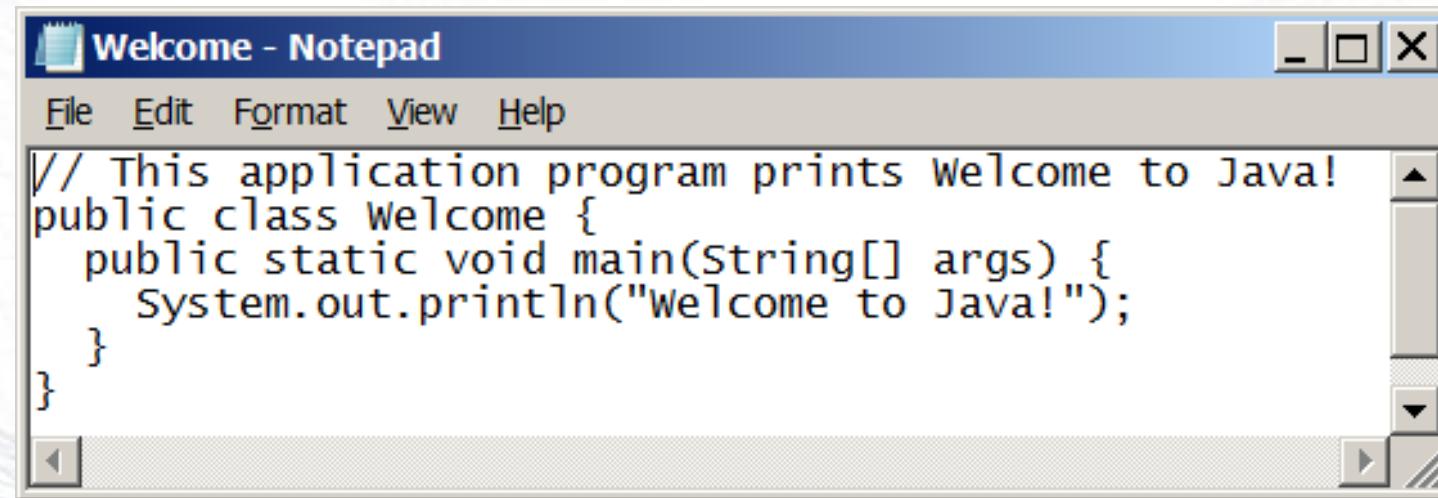
Note: Clicking the green button displays the source code with interactive animation. You can also run the code in a browser. Internet connection is needed for this button.

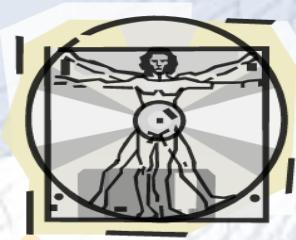
Note: Clicking the blue button runs the code from Windows. If you cannot run the buttons, see



Creating and Editing Using NotePad

To use NotePad, type
notepad Welcome.java
from the DOS prompt.





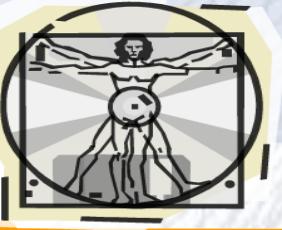
Creating and Editing Using WordPad

To use WordPad, type
write Welcome.java
from the DOS prompt.



```
// This application program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

Creating, Compiling, and Running Programs



```
Welcome - Notepad
File Edit Format View Help
// This application program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

Source code (developed by the programmer)

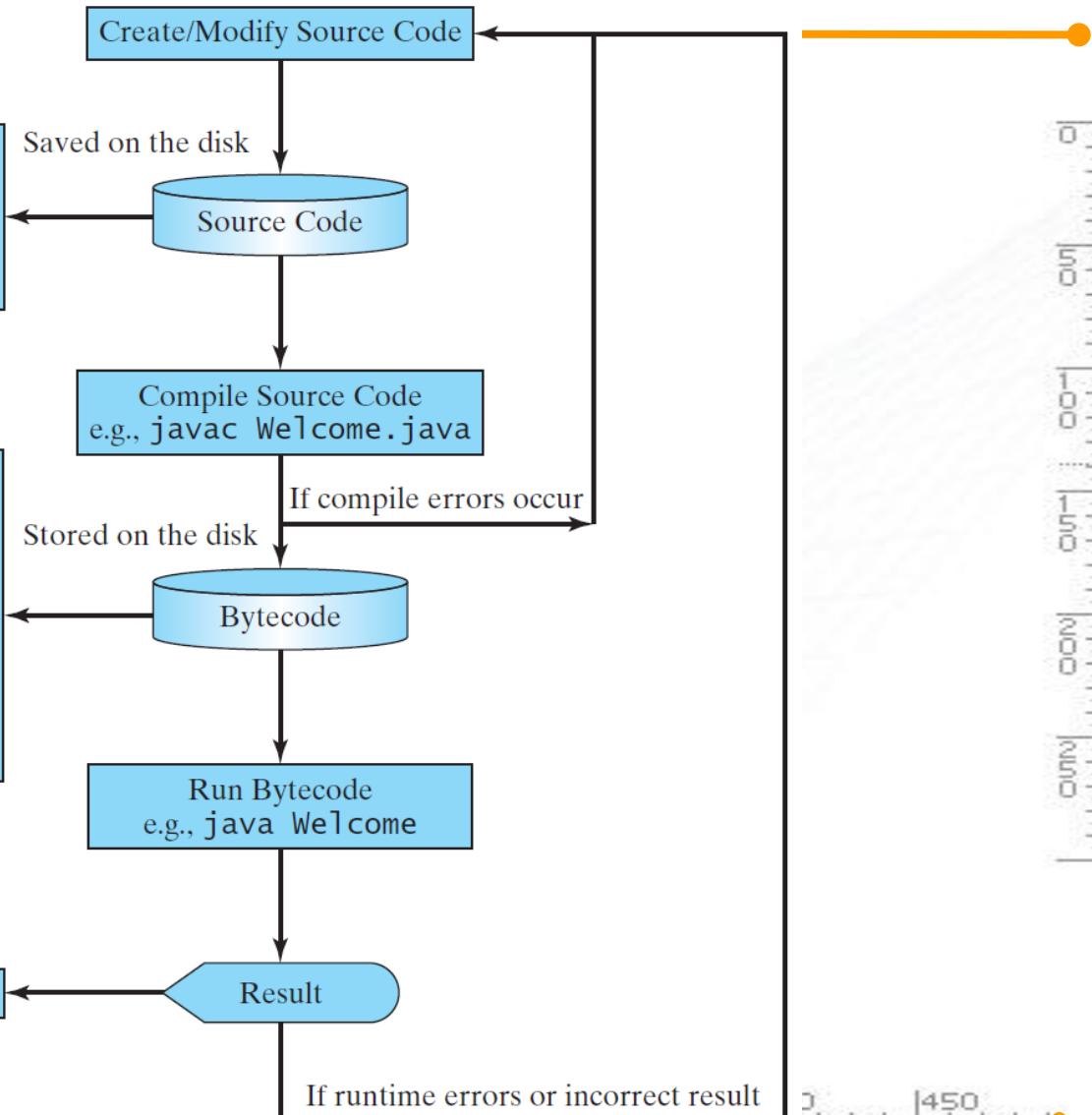
```
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

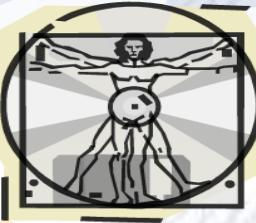
Bytecode (generated by the compiler for JVM to read and interpret)

```
...
Method Welcome()
0  aload_0
...
Method void main(java.lang.String[])
0  getstatic #2 ...
3  ldc #3 <String "Welcome to Java!">
5  invokevirtual #4 ...
8  return
```

“Welcome to Java” is displayed on the console

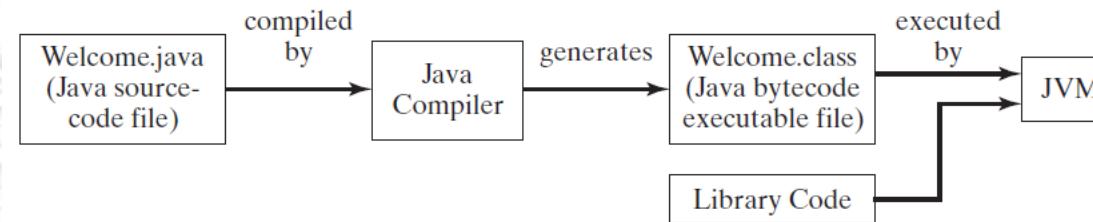
```
Welcome to Java!
```



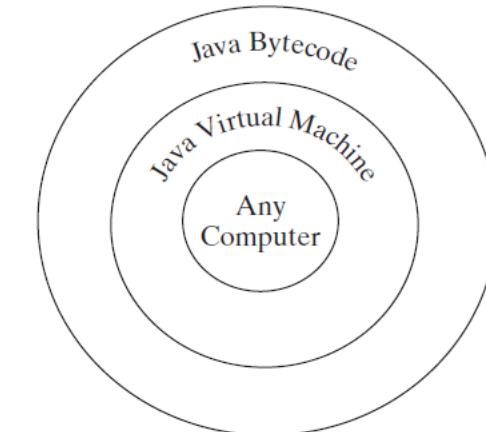


Compiling Java Source Code

You can port a source program to any machine with appropriate compilers. The source program must be recompiled, however, because the object program can only run on a specific machine. Nowadays computers are networked to work together. Java was designed to run object programs on any platform. With Java, you write the program once, and compile the source program into a special type of object code, known as *bytecode*. The bytecode can then run on any computer with a Java Virtual Machine, as shown below. Java Virtual Machine is a software that interprets Java bytecode.



(a)



(b)

Trace a Program Execution

Enter main method

```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

Trace a Program Execution

Execute statement

```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

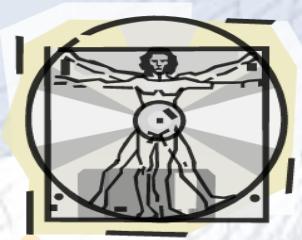
Trace a Program Execution

```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```



print a message to the
console

0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700
4800
4900
5000
5100
5200
5300
5400
5500
5600
5700
5800
5900
6000
6100
6200
6300
6400
6500
6600
6700
6800
6900
7000
7100
7200
7300
7400
7500
7600
7700
7800
7900
8000
8100
8200
8300
8400
8500
8600
8700
8800
8900
9000
9100
9200
9300
9400
9500
9600
9700
9800
9900
10000
10100
10200
10300
10400
10500
10600
10700
10800
10900
11000
11100
11200
11300
11400
11500
11600
11700
11800
11900
12000
12100
12200
12300
12400
12500
12600
12700
12800
12900
13000
13100
13200
13300
13400
13500
13600
13700
13800
13900
14000
14100
14200
14300
14400
14500
14600
14700
14800
14900
15000
15100
15200
15300
15400
15500
15600
15700
15800
15900
16000
16100
16200
16300
16400
16500
16600
16700
16800
16900
17000
17100
17200
17300
17400
17500
17600
17700
17800
17900
18000
18100
18200
18300
18400
18500
18600
18700
18800
18900
19000
19100
19200
19300
19400
19500
19600
19700
19800
19900
20000
20100
20200
20300
20400
20500
20600
20700
20800
20900
21000
21100
21200
21300
21400
21500
21600
21700
21800
21900
22000
22100
22200
22300
22400
22500
22600
22700
22800
22900
23000
23100
23200
23300
23400
23500
23600
23700
23800
23900
24000
24100
24200
24300
24400
24500
24600
24700
24800
24900
25000
25100
25200
25300
25400
25500
25600
25700
25800
25900
26000
26100
26200
26300
26400
26500
26600
26700
26800
26900
27000
27100
27200
27300
27400
27500
27600
27700
27800
27900
28000
28100
28200
28300
28400
28500
28600
28700
28800
28900
29000
29100
29200
29300
29400
29500
29600
29700
29800
29900
30000
30100
30200
30300
30400
30500
30600
30700
30800
30900
31000
31100
31200
31300
31400
31500
31600
31700
31800
31900
32000
32100
32200
32300
32400
32500
32600
32700
32800
32900
33000
33100
33200
33300
33400
33500
33600
33700
33800
33900
34000
34100
34200
34300
34400
34500
34600
34700
34800
34900
35000
35100
35200
35300
35400
35500
35600
35700
35800
35900
36000
36100
36200
36300
36400
36500
36600
36700
36800
36900
37000
37100
37200
37300
37400
37500
37600
37700
37800
37900
38000
38100
38200
38300
38400
38500
38600
38700
38800
38900
39000
39100
39200
39300
39400
39500
39600
39700
39800
39900
40000
40100
40200
40300
40400
40500
40600
40700
40800
40900
41000
41100
41200
41300
41400
41500
41600
41700
41800
41900
42000
42100
42200
42300
42400
42500
42600
42700
42800
42900
43000
43100
43200
43300
43400
43500
43600
43700
43800
43900
44000
44100
44200
44300
44400
44500
44600
44700
44800
44900
45000
45100
45200
45300
45400
45500
45600
45700
45800
45900
46000
46100
46200
46300
46400
46500
46600
46700
46800
46900
47000
47100
47200
47300
47400
47500
47600
47700
47800
47900
48000
48100
48200
48300
48400
48500
48600
48700
48800
48900
49000
49100
49200
49300
49400
49500
49600
49700
49800
49900
50000
50100
50200
50300
50400
50500
50600
50700
50800
50900
51000
51100
51200
51300
51400
51500
51600
51700
51800
51900
52000
52100
52200
52300
52400
52500
52600
52700
52800
52900
53000
53100
53200
53300
53400
53500
53600
53700
53800
53900
54000
54100
54200
54300
54400
54500
54600
54700
54800
54900
55000
55100
55200
55300
55400
55500
55600
55700
55800
55900
56000
56100
56200
56300
56400
56500
56600
56700
56800
56900
57000
57100
57200
57300
57400
57500
57600
57700
57800
57900
58000
58100
58200
58300
58400
58500
58600
58700
58800
58900
59000
59100
59200
59300
59400
59500
59600
59700
59800
59900
60000
60100
60200
60300
60400
60500
60600
60700
60800
60900
61000
61100
61200
61300
61400
61500
61600
61700
61800
61900
62000
62100
62200
62300
62400
62500
62600
62700
62800
62900
63000
63100
63200
63300
63400
63500
63600
63700
63800
63900
64000
64100
64200
64300
64400
64500
64600
64700
64800
64900
65000
65100
65200
65300
65400
65500
65600
65700
65800
65900
66000
66100
66200
66300
66400
66500
66600
66700
66800
66900
67000
67100
67200
67300
67400
67500
67600
67700
67800
67900
68000
68100
68200
68300
68400
68500
68600
68700
68800
68900
69000
69100
69200
69300
69400
69500
69600
69700
69800
69900
70000
70100
70200
70300
70400
70500
70600
70700
70800
70900
71000
71100
71200
71300
71400
71500
71600
71700
71800
71900
72000
72100
72200
72300
72400
72500
72600
72700
72800
72900
73000
73100
73200
73300
73400
73500
73600
73700
73800
73900
74000
74100
74200
74300
74400
74500
74600
74700
74800
74900
75000
75100
75200
75300
75400
75500
75600
75700
75800
75900
76000
76100
76200
76300
76400
76500
76600
76700
76800
76900
77000
77100
77200
77300
77400
77500
77600
77700
77800
77900
78000
78100
78200
78300
78400
78500
78600
78700
78800
78900
79000
79100
79200
79300
79400
79500
79600
79700
79800
79900
80000
80100
80200
80300
80400
80500
80600
80700
80800
80900
81000
81100
81200
81300
81400
81500
81600
81700
81800
81900
82000
82100
82200
82300
82400
82500
82600
82700
82800
82900
83000
83100
83200
83300
83400
83500
83600
83700
83800
83900
84000
84100
84200
84300
84400
84500
84600
84700
84800
84900
85000
85100
85200
85300
85400
85500
85600
85700
85800
85900
86000
86100
86200
86300
86400
86500
86600
86700
86800
86900
87000
87100
87200
87300
87400
87500
87600
87700
87800
87900
88000
88100
88200
88300
88400
88500
88600
88700
88800
88900
89000
89100
89200
89300
89400
89500
89600
89700
89800
89900
90000
90100
90200
90300
90400
90500
90600
90700
90800
90900
91000
91100
91200
91300
91400
91500
91600
91700
91800
91900
92000
92100
92200
92300
92400
92500
92600
92700
92800
92900
93000
93100
93200
93300
93400
93500
93600
93700
93800
93900
94000
94100
94200
94300
94400
94500
94600
94700
94800
94900
95000
95100
95200
95300
95400
95500
95600
95700
95800
95900
96000
96100
96200
96300
96400
96500
96600
96700
96800
96900
97000
97100
97200
97300
97400
97500
97600
97700
97800
97900
98000
98100
98200
98300
98400
98500
98600
98700
98800
98900
99000
99100
99200
99300
99400
99500
99600
99700
99800
99900
100000



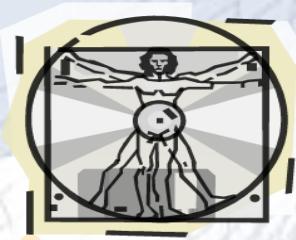
Two More Simple Examples

WelcomeWithThreeMessages

Run

ComputeExpression

Run



Supplements on the Companion Website

- See Supplement I.B for installing and configuring JDK
- See Supplement I.C for compiling and running Java from the command window for details

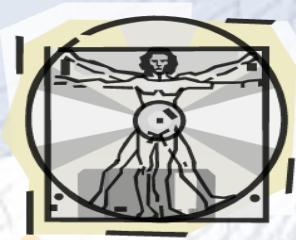
www.pearsonhighered.com/liang



Compiling and Running Java from the Command Window

- Set path to JDK bin directory
 - set path=c:\Program Files\java\jdk1.8.0\bin
- Set classpath to include the current directory
 - set classpath=.
- Compile
 - javac Welcome.java
- Run
 - java Welcome

```
C:\book>javac Welcome.java
C:\book>dir Welcome.* 
Volume in drive C has no label.
Volume Serial Number is 9CB6-16F1
Directory of C:\book
07/31/2003  03:32p           424 Welcome.class
06/20/2003  07:39p           119 Welcome.java
                           2 File(s)      543 bytes
                           0 Dir(s)  21,700,853,760 bytes free
C:\book>java Welcome
Welcome to Java!
C:\book>
```

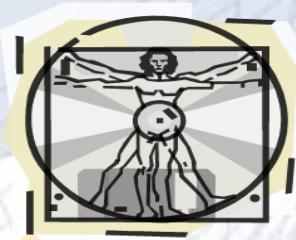


Compiling and Running Java from TextPad

- See Supplement II.A on the Website for details

The screenshot shows the TPad application window. The title bar reads "TextPad - [C:\MyPrograms\Welcome.java]". The menu bar includes File, Edit, Search, View, Tools, Macros, Configure, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Find. On the left, there's a sidebar with a file list showing "Welcome.java" and an "ANSI Characters" panel displaying a table of character codes 33 through 35 with their corresponding symbols (!, ", #). The main editor area contains the following Java code:

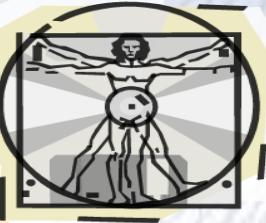
```
// Welcome.java: This application program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```



Anatomy of a Java Program

- Class name
- Main method
- Statements
- Statement terminator
- Reserved words
- Comments
- Blocks

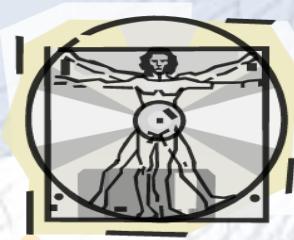




Class Name

Every Java program must have at least one class. Each class has a name. By convention, class names start with an uppercase letter. In this example, the class name is Welcome.

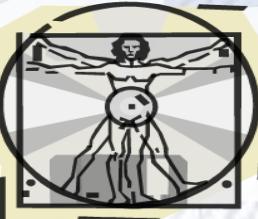
```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```



Main Method

Line 2 defines the main method. In order to run a class, the class must contain a method named main. The program is executed from the main method.

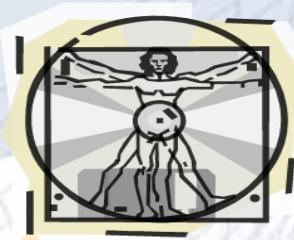
```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```



Statement

A statement represents an action or a sequence of actions. The statement `System.out.println("Welcome to Java!")` in the program in Listing 1.1 is a statement to display the greeting "Welcome to Java!".

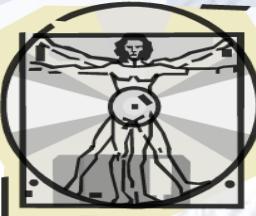
```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```



Statement Terminator

Every statement in Java ends with a semicolon (;).

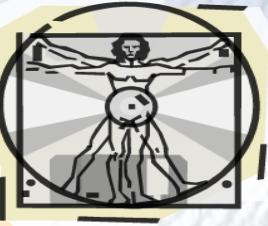
```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!"); }
}
```



Reserved words

Reserved words or keywords are words that have a specific meaning to the compiler and cannot be used for other purposes in the program. For example, when the compiler sees the word class, it understands that the word after class is the name for the class.

```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```



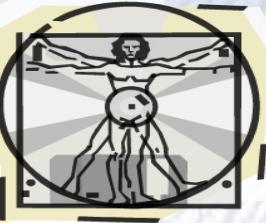
Blocks

A pair of braces in a program forms a block that groups components of a program.

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

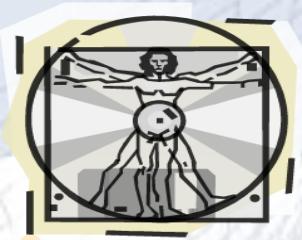
Class block

Method block



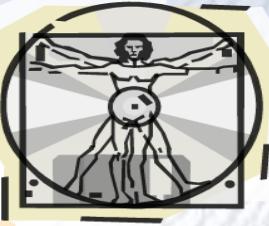
Special Symbols

Character Name	Description	
{ }	Opening and closing braces	Denotes a block to enclose statements.
()	Opening and closing parentheses	Used with methods.
[]	Opening and closing brackets	Denotes an array.
//	Double slashes	Precedes a comment line.
" "	Opening and closing quotation marks	Enclosing a string (i.e., sequence of characters).
;	Semicolon	Marks the end of a statement.



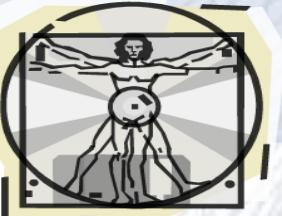
{ ... }

```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

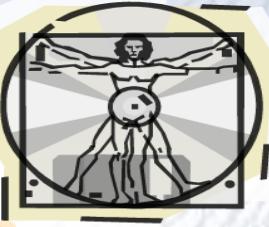


(...)

```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

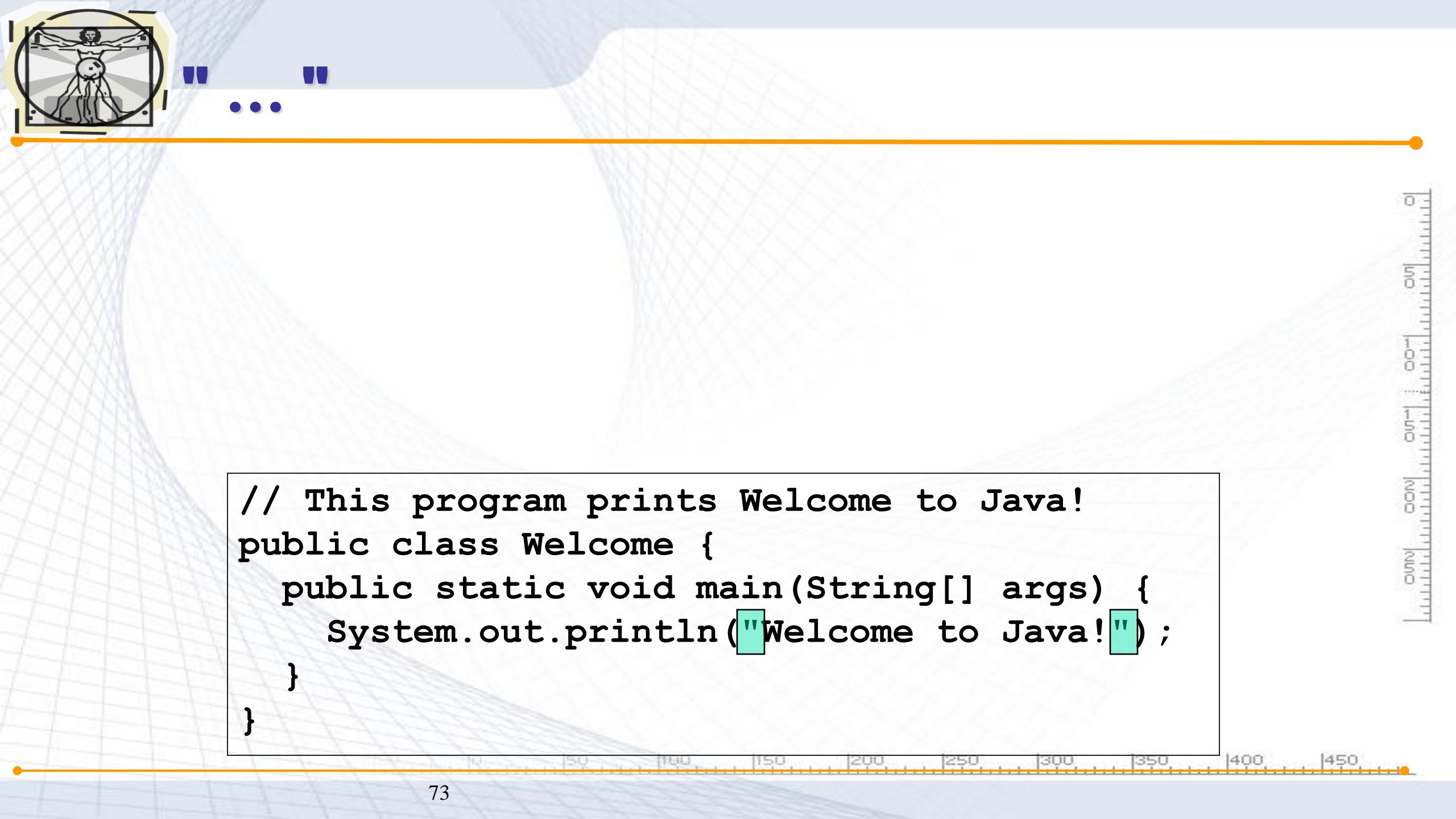


```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

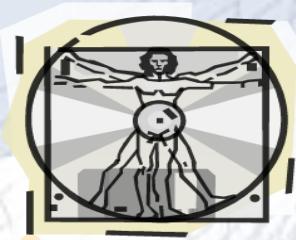


0
100
200
300
400
500
600
700
800
900

```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```



```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

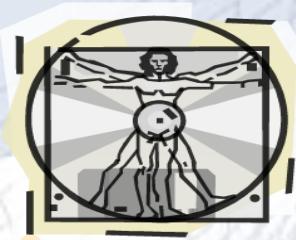


How do you reach the following output

```
public class WelcomeJava {  
    public static void main(String[] args) {  
        System.out.println(" ***");  
        System.out.println(" **");  
        System.out.println(" *");  
    }  
}
```

The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

**
*
The console tab also shows the application name and path: <terminated> WelcomeJava [Java Application] C:\Users\... .



How do you reach the following output

```
public class WelcomeJava {  
    public static void main(String[] args) {  
        System.out.println("*****");  
        System.out.println(" * * *");  
        System.out.println(" * *");  
        System.out.println(" * *");  
        System.out.println(" *");  
    }  
}
```

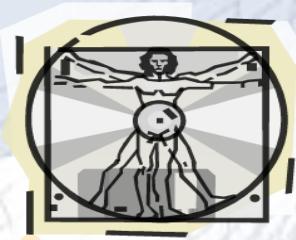
The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

* * *
* *
*
The vertical ruler on the right indicates the code length is approximately 200 characters.



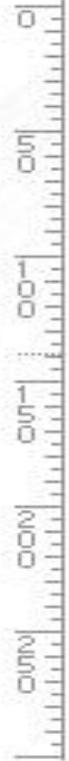
What will be the Output

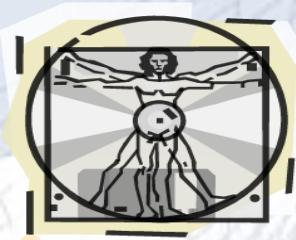
- `System.out.println(7);` 7
- `System.out.println("7");` 7
- `System.out.println("7+2");` 7+2
- `System.out.println(7+2);` 9
- `System.out.println("Kayseri"+89);` Kayseri89
- `System.out.println(89+"Kayseri");` 89Kayseri
- `System.out.println(89+2+"Kayseri");` 91Kayseri
- `System.out.println(89+2+"Kayseri"+89+2);` 91Kayseri892
- `System.out.println(89+2+"Kayseri"+(89+2));` 91Kayseri91



Programming Style and Documentation

- Appropriate Comments
- Naming Conventions
- Proper Indentation and Spacing Lines
- Block Styles

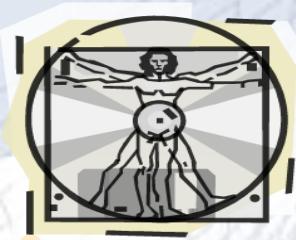




Appropriate Comments

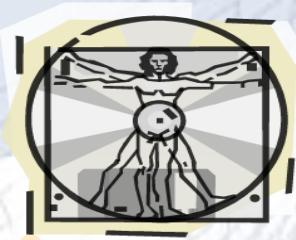
Include a summary at the beginning of the program to explain what the program does, its key features, its supporting data structures, and any unique techniques it uses.

Include your name, class section, instructor, date, and a brief description at the beginning of the program.



Naming Conventions

- Choose meaningful and descriptive names.
- Class names:
 - Capitalize the first letter of each word in the name. For example, the class name `ComputeExpression`.

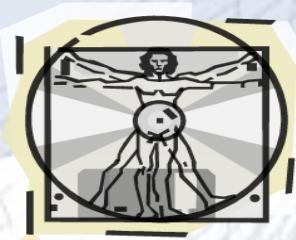


Proper Indentation and Spacing

- Indentation
 - Indent two spaces.

- Spacing
 - Use blank line to separate segments of the code.





Block Styles

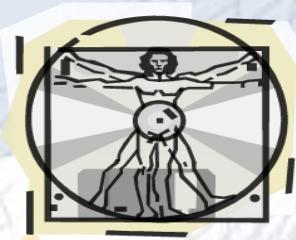
Use end-of-line style for braces.

*Next-line
style*

```
public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Block Styles");
    }
}
```

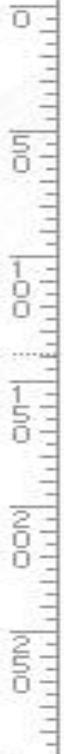
*End-of-line
style*

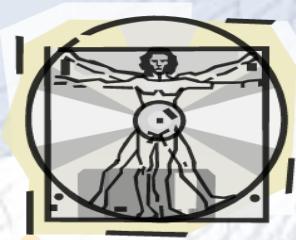
```
public class Test {
    public static void main(String[] args) {
        System.out.println("Block Styles");
    }
}
```



Programming Errors

- Syntax Errors
 - Detected by the compiler
- Runtime Errors
 - Causes the program to abort
- Logic Errors
 - Produces incorrect result



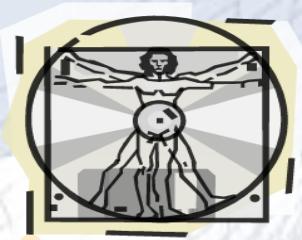


Syntax Errors

```
public class ShowSyntaxErrors {  
    public static main(String[] args) {  
        System.out.println("Welcome to Java");  
    }  
}
```

ShowSyntaxErrors

Run

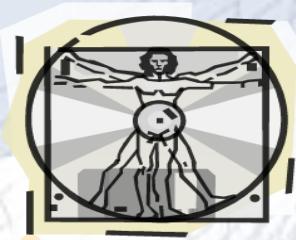


Runtime Errors

```
public class ShowRuntimeErrors {  
    public static void main(String[] args) {  
        System.out.println(1 / 0);  
    }  
}
```

ShowRuntimeErrors

Run



Logic Errors

```
public class ShowLogicErrors {  
    public static void main(String[] args) {  
        System.out.println("Celsius 35 is Fahrenheit degree ");  
        System.out.println((9 / 5) * 35 + 32);  
    }  
}
```

ShowLogicErrors

Run