Objectives

✦ To understand computer basics, programs, and operating systems (§§1.2–1.4).
✦ To describe the relationship between Java and the World Wide Web (§1.5).
✦ To understand the meaning of Java language specification, API, JDK, and IDE (§1.6).
✦ To write a simple Java program (§1.7).
✦ To display output on the console (§1.7).
✦ To explain the basic syntax of a Java program (§1.7).
✦ To create, compile, and run Java programs (§1.8).
✦ To use sound Java programming style and document programs properly (§1.9).
✦ To explain the differences between syntax errors, runtime errors, and logic errors (§1.10).
✦ To develop Java programs using NetBeans (§1.11).
✦ To develop Java programs using Eclipse (§1.12).
What is a Computer?

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

- A computer’s components are interconnected by a subsystem called a *bus*. You can think of a bus as a sort of system of roads running among the computer’s components; data and power travel along the bus from one part of the computer to another. In personal computers the bus is built into the computer’s *motherboard*, which is a circuit case that connects all of the parts of a computer together.
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.

<table>
<thead>
<tr>
<th>Memory address</th>
<th>Memory content</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>01001010</td>
</tr>
<tr>
<td>2001</td>
<td>01100001</td>
</tr>
<tr>
<td>2002</td>
<td>01110110</td>
</tr>
<tr>
<td>2003</td>
<td>01100001</td>
</tr>
<tr>
<td>2004</td>
<td>00000011</td>
</tr>
</tbody>
</table>

Encoding for character ‘J’
Encoding for character ‘a’
Encoding for character ‘v’
Encoding for character ‘a’
Encoding for number 3
### ASCII TABLE

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hx Oct</th>
<th>Char</th>
<th>Dec</th>
<th>Hx Oct</th>
<th>Html</th>
<th>Chr</th>
<th>Dec</th>
<th>Hx Oct</th>
<th>Html</th>
<th>Chr</th>
<th>Dec</th>
<th>Hx Oct</th>
<th>Html</th>
<th>Chr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>NUL (null)</td>
<td>32</td>
<td>20</td>
<td>040</td>
<td>#32; &quot;Space&quot;</td>
<td>64</td>
<td>40</td>
<td>100</td>
<td>#64; 8</td>
<td>96</td>
<td>60</td>
<td>140</td>
<td>#96; `</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>SOH (start of heading)</td>
<td>33</td>
<td>21</td>
<td>041</td>
<td>#33;</td>
<td>65</td>
<td>41</td>
<td>101</td>
<td>#65; A</td>
<td>97</td>
<td>61</td>
<td>141</td>
<td>#97; a</td>
</tr>
<tr>
<td>2</td>
<td>0002</td>
<td>STX (start of text)</td>
<td>34</td>
<td>22</td>
<td>042</td>
<td>#34; '</td>
<td>66</td>
<td>42</td>
<td>102</td>
<td>#66; B</td>
<td>98</td>
<td>62</td>
<td>142</td>
<td>#98; b</td>
</tr>
<tr>
<td>3</td>
<td>0003</td>
<td>ETX (end of text)</td>
<td>35</td>
<td>23</td>
<td>043</td>
<td>#35; ¦</td>
<td>67</td>
<td>43</td>
<td>103</td>
<td>#67; C</td>
<td>99</td>
<td>63</td>
<td>143</td>
<td>#99; c</td>
</tr>
<tr>
<td>4</td>
<td>0004</td>
<td>EOT (end of transmission)</td>
<td>36</td>
<td>24</td>
<td>044</td>
<td>#36; $</td>
<td>68</td>
<td>44</td>
<td>104</td>
<td>#68; D</td>
<td>100</td>
<td>64</td>
<td>144</td>
<td>#100; d</td>
</tr>
<tr>
<td>5</td>
<td>0005</td>
<td>ENQ (enquiry)</td>
<td>37</td>
<td>25</td>
<td>045</td>
<td>#37; %</td>
<td>69</td>
<td>45</td>
<td>105</td>
<td>#69; E</td>
<td>101</td>
<td>65</td>
<td>145</td>
<td>#101; e</td>
</tr>
<tr>
<td>6</td>
<td>0006</td>
<td>ACK (acknowledge)</td>
<td>38</td>
<td>26</td>
<td>046</td>
<td>#38; &amp;</td>
<td>70</td>
<td>46</td>
<td>106</td>
<td>#70; F</td>
<td>102</td>
<td>66</td>
<td>146</td>
<td>#102; f</td>
</tr>
<tr>
<td>7</td>
<td>0007</td>
<td>BEL (bell)</td>
<td>39</td>
<td>27</td>
<td>047</td>
<td>#39; '</td>
<td>71</td>
<td>47</td>
<td>107</td>
<td>#71; G</td>
<td>103</td>
<td>67</td>
<td>147</td>
<td>#103; g</td>
</tr>
<tr>
<td>8</td>
<td>010</td>
<td>BS (backspace)</td>
<td>40</td>
<td>28</td>
<td>050</td>
<td>#40; (</td>
<td>72</td>
<td>48</td>
<td>110</td>
<td>#72; H</td>
<td>104</td>
<td>68</td>
<td>150</td>
<td>#104; h</td>
</tr>
<tr>
<td>9</td>
<td>011</td>
<td>TAB (horizontal tab)</td>
<td>41</td>
<td>29</td>
<td>051</td>
<td>#41; )</td>
<td>73</td>
<td>49</td>
<td>111</td>
<td>#73; I</td>
<td>105</td>
<td>69</td>
<td>151</td>
<td>#105; i</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>LF (NL line feed, new line)</td>
<td>42</td>
<td>2A</td>
<td>052</td>
<td>#42; ^</td>
<td>74</td>
<td>4A</td>
<td>112</td>
<td>#74; J</td>
<td>106</td>
<td>6A</td>
<td>152</td>
<td>#106; j</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>VT (vertical tab)</td>
<td>43</td>
<td>2B</td>
<td>053</td>
<td>#43; +</td>
<td>75</td>
<td>4B</td>
<td>113</td>
<td>#75; K</td>
<td>107</td>
<td>6B</td>
<td>153</td>
<td>#107; k</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>FF (NP form feed, new page)</td>
<td>44</td>
<td>2C</td>
<td>054</td>
<td>#44;</td>
<td>76</td>
<td>4C</td>
<td>114</td>
<td>#76; L</td>
<td>108</td>
<td>6C</td>
<td>154</td>
<td>#108; l</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>CR (carriage return)</td>
<td>45</td>
<td>2D</td>
<td>055</td>
<td>#45; -</td>
<td>77</td>
<td>4D</td>
<td>115</td>
<td>#77; M</td>
<td>109</td>
<td>6D</td>
<td>155</td>
<td>#109; m</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>SO (shift out)</td>
<td>46</td>
<td>2E</td>
<td>056</td>
<td>#46;</td>
<td>78</td>
<td>4E</td>
<td>116</td>
<td>#78; N</td>
<td>110</td>
<td>6E</td>
<td>156</td>
<td>#110; n</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>SI (shift in)</td>
<td>47</td>
<td>2F</td>
<td>057</td>
<td>#47; /</td>
<td>79</td>
<td>4F</td>
<td>117</td>
<td>#79; O</td>
<td>111</td>
<td>6F</td>
<td>157</td>
<td>#111; o</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>DLE (data link escape)</td>
<td>48</td>
<td>30</td>
<td>060</td>
<td>#48; 0</td>
<td>80</td>
<td>50</td>
<td>120</td>
<td>#80; P</td>
<td>112</td>
<td>70</td>
<td>160</td>
<td>#112; p</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>DC1 (device control 1)</td>
<td>49</td>
<td>31</td>
<td>061</td>
<td>#49; 1</td>
<td>81</td>
<td>51</td>
<td>121</td>
<td>#81; Q</td>
<td>113</td>
<td>71</td>
<td>161</td>
<td>#113; q</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>DC2 (device control 2)</td>
<td>50</td>
<td>32</td>
<td>062</td>
<td>#50; 2</td>
<td>82</td>
<td>52</td>
<td>122</td>
<td>#82; R</td>
<td>114</td>
<td>72</td>
<td>162</td>
<td>#114; r</td>
</tr>
<tr>
<td>19</td>
<td>10</td>
<td>DC3 (device control 3)</td>
<td>51</td>
<td>33</td>
<td>063</td>
<td>#51; 3</td>
<td>83</td>
<td>53</td>
<td>123</td>
<td>#83; S</td>
<td>115</td>
<td>73</td>
<td>163</td>
<td>#115; s</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>DC4 (device control 4)</td>
<td>52</td>
<td>34</td>
<td>064</td>
<td>#52; 4</td>
<td>84</td>
<td>54</td>
<td>124</td>
<td>#84; T</td>
<td>116</td>
<td>74</td>
<td>164</td>
<td>#116; t</td>
</tr>
<tr>
<td>21</td>
<td>10</td>
<td>NAK (negative acknowledge)</td>
<td>53</td>
<td>35</td>
<td>065</td>
<td>#53; 5</td>
<td>85</td>
<td>55</td>
<td>125</td>
<td>#85; U</td>
<td>117</td>
<td>75</td>
<td>165</td>
<td>#117; u</td>
</tr>
<tr>
<td>22</td>
<td>10</td>
<td>SYM (synchronous idle)</td>
<td>54</td>
<td>36</td>
<td>066</td>
<td>#54; 6</td>
<td>86</td>
<td>56</td>
<td>126</td>
<td>#86; V</td>
<td>118</td>
<td>76</td>
<td>166</td>
<td>#118; v</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>ETB (end of trans. block)</td>
<td>55</td>
<td>37</td>
<td>067</td>
<td>#55; 7</td>
<td>87</td>
<td>57</td>
<td>127</td>
<td>#87; W</td>
<td>119</td>
<td>77</td>
<td>167</td>
<td>#119; w</td>
</tr>
<tr>
<td>24</td>
<td>10</td>
<td>CAN (cancel)</td>
<td>56</td>
<td>38</td>
<td>070</td>
<td>#56; 8</td>
<td>88</td>
<td>58</td>
<td>130</td>
<td>#88; X</td>
<td>120</td>
<td>78</td>
<td>170</td>
<td>#120; x</td>
</tr>
<tr>
<td>25</td>
<td>10</td>
<td>EM (end of medium)</td>
<td>57</td>
<td>39</td>
<td>071</td>
<td>#57; 9</td>
<td>89</td>
<td>59</td>
<td>131</td>
<td>#89; Y</td>
<td>121</td>
<td>79</td>
<td>171</td>
<td>#121; y</td>
</tr>
<tr>
<td>26</td>
<td>10</td>
<td>SUB (substitute)</td>
<td>58</td>
<td>3A</td>
<td>072</td>
<td>#58; :</td>
<td>90</td>
<td>5A</td>
<td>132</td>
<td>#90; Z</td>
<td>122</td>
<td>7A</td>
<td>172</td>
<td>#122; z</td>
</tr>
<tr>
<td>27</td>
<td>10</td>
<td>ESC (escape)</td>
<td>59</td>
<td>3B</td>
<td>073</td>
<td>#59;</td>
<td>91</td>
<td>5B</td>
<td>133</td>
<td>#91; [</td>
<td>123</td>
<td>7B</td>
<td>173</td>
<td>#123; {</td>
</tr>
<tr>
<td>28</td>
<td>10</td>
<td>FS (file separator)</td>
<td>60</td>
<td>3C</td>
<td>074</td>
<td>#60; &lt;</td>
<td>92</td>
<td>5C</td>
<td>134</td>
<td>#92; \</td>
<td>124</td>
<td>7C</td>
<td>174</td>
<td>#124;</td>
</tr>
<tr>
<td>29</td>
<td>10</td>
<td>GS (group separator)</td>
<td>61</td>
<td>3D</td>
<td>075</td>
<td>#61; =</td>
<td>93</td>
<td>5D</td>
<td>135</td>
<td>#93; ]</td>
<td>125</td>
<td>7D</td>
<td>175</td>
<td>#125; }</td>
</tr>
<tr>
<td>30</td>
<td>10</td>
<td>RS (record separator)</td>
<td>62</td>
<td>3E</td>
<td>076</td>
<td>#62; &gt;</td>
<td>94</td>
<td>5E</td>
<td>136</td>
<td>#94; ^</td>
<td>126</td>
<td>7E</td>
<td>176</td>
<td>#126; ~</td>
</tr>
<tr>
<td>31</td>
<td>10</td>
<td>US (unit separator)</td>
<td>63</td>
<td>3F</td>
<td>077</td>
<td>#63; ?</td>
<td>95</td>
<td>5F</td>
<td>137</td>
<td>#95; _</td>
<td>127</td>
<td>7F</td>
<td>177</td>
<td>#127; DEL</td>
</tr>
</tbody>
</table>

Source: www.LookupTables.com
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
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:

\[
\text{area} = 5 \times 5 \times 3.1415;
\]
## Popular High-Level Languages

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

```java
data = 5 * 5 * 3.1415;
... 
```
Operating Systems

The *operating system* (OS) is a program that manages and controls a computer’s activities. The popular operating systems for general-purpose computers are Microsoft Windows, Mac OS, and Linux. Application programs, such as a Web browser or a word processor, cannot run unless an operating system is installed and 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.

✦ Java is a general purpose programming language.
✦ Java is the Internet programming language.
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:

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) a. k. a. JDK 5 or Java 5
✦ JDK 1.6 (2006) a. k. a. JDK 6 or Java 6
✦ JDK 1.7 (2011) a. k. a. JDK 7 or Java 7
✦ JDK 1.8 (2014) a. k. a. JDK 8 or Java 8
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.
Popular Java IDEs

- NetBeans
- Eclipse
A Simple Java Program

Listing 1.1

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

Source code (developed by the programmer)

```java
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!
```

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.
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.
// 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

Enter main method
Trace a Program Execution

```java
// 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!");
    }
}
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
Compiling and Running Java from TextPad

✧ See Supplement II.A on the Website for details