

AP Computer Science A

Course Map

This course is designed to prepare students to take the Advanced Placement (AP*) exams conducted by the College Board. The course is based on a comprehensive AP* syllabus that is rigorously aligned to the curricular design and test requirements of the College Board.

[AP Course Audit Standards](#)

Semester A:

Unit 1: Introduction to Computer Programming

Unit 1:

By the end of this unit, you will be able to do the following:

- Describe and explain the basic concepts of computer programming.
- Compile and run a simple Java program.
- Learn to identify, describe, and employ Java variables and data types.
- Describe and use unary and arithmetic operators, perform arithmetic calculations, and use the Math class.
- Describe and use relational and logical operators.
- Describe and use decision-making statements (if, if...else, nested if).
- Use different types of loop statements (while, for, nested) and implement algorithms.

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
Computer Science	In this lesson, you will	■ Computer Science: ○ Discussion	■ Dictionary	■ Read the instructions for this self-checked activity. Type in your response to each question,	■ Lesson Activities:

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
	describe and explain the basic concepts of computer programming.	○ Tutorial ○ Unit Activity: Introduction to Computer Programming	■ Scientific Calculator ■ Click to Speak ■ Translate ■ Highlighter ■ Discussion Board ■ Slide Narration ■ Glossary	and check your answers. At the end of the activity, write a brief evaluation of your work. ■ You looked at how computers evolved over time and the parts of the present-day computer. In this activity, you will ponder if there would a difference in our lives without computers. You will also identify a recent problem that a computer can solve. ■ Speculate on how life in the present day would be different if digital electronic computers did not exist. ■ Identify a recent problem that you think that computers could help solve. Explain how computers would be used to help solve the problem. ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. ■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ In this activity, you will convert binary, hexadecimal, and octal numbers to decimal integers, and convert octal and hexadecimal	Self-Evaluations ■ Mastery Test ■ Unit 1 Post Test ■ AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				numbers to binary. Please do all conversions by hand. Show your work. - Convert the following binary numbers to decimal integers: - Convert 11_2 to decimal integer. - Convert 1011_2 to decimal integer. - Convert the following octal numbers to decimal integers. - Convert 104_8 to decimal integer. - Convert 45_8 to decimal integer. - Convert the following octal numbers to binary integers. - Convert 22_8 to binary integer. - Convert 74_8 to binary integer. - Convert the following hexadecimal numbers to decimal integers. - Convert 12_{16} to decimal integer. - Convert $5E_{16}$ to decimal integer. - Convert the following hexadecimal numbers to binary numbers. - Convert $E4A0_{16}$ to binary number. - Convert $F9_{16}$ to binary number.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - Imagine that you are in charge of the production of a new piece of application software. In this activity, you will describe your requirements for the software. - Describe the software and what it will be used for. - List and explain your requirements for the software. - Describe how your software could be verified for correct operation. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Java Basics	In this lesson, you will compile and run a simple	- Java Basics: - Discussion - Tutorial	- Dictionary - Scientific Calculator	Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the	Lesson Activities:

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
	Java program.	Unit Activity: Introduction to Computer Programming	- Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	activity, write a brief evaluation of your work. - In the lesson you learned the two methods commonly used to create, compile, and run Java programs. Now, use both methods to compile and run the Hello World program. State whether both programs compiled and ran successfully. Describe which method you prefer to use for creating, compiling, and running Java programs in the future and why you prefer that method. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will use the syntax you learned to modify the Hello World program.	- Self-Evaluations - Mastery Test - Unit 1 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        // Prints a message to the console
        System.out.println("Hello, World!");
    }
}
```

- Compile the **HelloWorld** program (shown above) three different times after attempting each of the following. Report the error message received each time.
 - Remove the semicolon from the sixth line of the program.
 - Replace the semicolon removed in the previous step and change **println** to **printlm** on the same line.
 - Replace the **println** from the previous step. Then, introduce an error of your own that will produce a different compile-time error message. Describe the syntax error that you introduced.
- Modify the HelloWorld program to display a different message. Add single line and multi-line comments that relate to the message.
- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You have learned about the different ways you can use <code>println</code> and <code>print</code> in Java to produce different output. In this activity you will output text in a specific format. - Write a Java program that exactly reproduces the following output: <pre> A thing is a thing not what is said of a thing </pre> - Write a Java program that exactly reproduces the following output (hint: the escape sequences <code>\</code> and <code>\\</code> are needed to print the <code>"</code> and <code>\</code> characters):	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
					
				How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Variables and Data Types	In this lesson, you will learn to identify, describe, and employ Java variables and data types.	- Variables and Data Types: - Discussion - Tutorial - Unit Activity: Introduction to Computer Programming	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You have learned the different rules that can be used when writing different identifiers. You have also learned the different situations that use different types of identifiers. In this activity, you will use different identifiers in different situations. - Determine if the following identifiers are valid. If they are invalid, state the reason. For example, the identifier min value is invalid as it uses a whitespace:	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 1 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				○ road ○ mid_point ○ 8ball ○ break ○ _log_value ○ EVERY1 ○ small@int ○ _log ○ NewNumber ○ \$varA ○ Class ■ In the following situations, determine a valid identifier following standard naming conventions. For example, the total amount of a grocery bill could use the identifier groceryBill. ○ a variable representing the force applied to a crankshaft ○ a variable representing the number of images ○ a class representing delivery trucks ○ a constant representing the golden ratio ○ a method reporting annual earnings ■ Determine the probable uses for the following list of identifiers. They could be variables or methods, constants, or classes. For example, FARADAY would be used as a constant. ○ tension	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				○ Item ○ super ○ \$value ○ IMPORT ○ StaffWorker ○ 4tune ○ newCharacter ○ MAX_NUM ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. ■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ In this activity, you will use different data types in different situations. ■ Given the following declaration statements, find the error in each statement. Specify why the error occurs in each statement. ■ For example, char newChar = "H"; would return an error because this is a character data type and not a string. ○ int gValue = 9.8; ○ char messageNew = 'Hello';	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				○ float widthMeasure = 5.5; ○ boolean facChoice = 0; ○ char labelChar = S; ■ Write appropriate declaration statements for each of the following situations. ■ For example, an integer representing the number of floors in a building would be int numFloors; ○ a boolean representing the presence of a valid data file ○ a float constant storing the value of pi as 3.142 ○ two integers representing the age and grade of a student, initialized with values 17 and 12 ○ a double representing the wavelength of a laser with value 292 ○ an integer constant representing maximum of 100 iterations ○ a boolean representing the completion of an algorithm set to false ■ Write a program to determine the minimum and maximum possible values of the following types: byte, short, long, int, float, and double. For each value, the program should produce output similar to "The minimum value of type byte is -128". (Hint: review the use of System.out.print and System.out.println from	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				the previous lesson.) - Calculate the amount of memory needed for the following: 1024 variables of type double 40 variables of type int 1048576 variables of type long 2048 variables of type byte - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You have learned about the different ways you can use println and print in Java to produce different outputs. In this activity you will output text in a specific format. - Write the output produced by the following code. Also write a table that tabulates the values of a, b, and c after each assignment statement. <pre>int a = 1, b = 3, c = 4; System.out.printf("%d %d %d\n",a,b,c);</pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> b = 5; a = 2; c = b; a = c; System.out.printf("%d %d %d\n",a,b,c); b = a; a = 9; c = a; System.out.printf("%d %d %d\n",a,b,c); </pre> - Write a program that assigns the minimum values of the byte, short, and int types to int variables byteMin, shortMin, and intMin. The programs should produce the following output: "The values of byteMin, shortMin and intMin are X, Y, and Z, respectively.", where X, Y, and Z should be replaced with the actual values of the variables. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Arithmetic Operators and the Math Class	In this lesson, you will describe and use unary and arithmetic operators,	- Arithmetic Operators and the Math Class: - Discussion - Tutorial - Unit Activity: Introduction to	- Dictionary - Scientific Calculator - Click to Speak - Translate	Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.	- Lesson Activities: Self-Evaluations - Mastery Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
	perform arithmetic calculations, and use the Math class.	Computer Programming	- Highlighter - Discussion Board - Slide Narration - Glossary	- You looked at different types of operators in Java. In this activity, you will convert algebraic expressions to Java expressions. You will also evaluate algebraic expressions. - Convert the following algebraic expressions to Java expressions. $3y + 7$ $(2)(3) + 4r$ $\frac{1}{4}(3y)$ $7.8y - \frac{z}{2}$ - Evaluate the following expressions. Make sure that each answer has the correct type. $4 / 2$ $\frac{6}{8}$ $\frac{12.0}{5}$ $\frac{13}{11}$ $5 \% 6$ $17 \% 14$ $27 \% 9$ $19 \% 5$ - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a	- Unit 1 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

brief evaluation of your work below. Note what you learned and what challenged you.

- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.
- You came across several methods from the Math class. In this activity, you will convert mathematical expressions to Java expressions. You will also write a program to produce a random integer.
- Convert the following mathematical expressions to Java expressions.

- $\cot\left(\frac{\pi}{11}\right)$
- $\tan(44^\circ)$
- $e^{-6.5}$
- x^7
- $\log_{10}(2.5) - x^{7+y}$
- $|15x|$
- $\sqrt{45}$
- $\cos^2 x$

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Write a program to produce a random integer ranging from 1 to 500. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You looked at evaluation of expressions based on precedence. In this activity, you will evaluate expressions and show their output based on their data type. You will also write a program that generates a random integer and displays it in octal notation. - Evaluate the following expressions. Make sure that each answer has the correct type. $8.0 + 6 / 4 + 1$ $2.0 + 4 / 6 * 6$ $(1.0 + 2.0) / (3.0 + 3.0)$ $4 * 3 \% 5 + 2$	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Write a program that generates a random integer from 1 to 500 and writes it in octal notation using the arithmetic operators discussed in this lesson. (Hint: Use <code>System.out.print</code> to print one digit at a time. Remember that, for a three-digit octal number, the first digit represents the 8^2 (i.e., 64's) place, the second represents the 8^1 (i.e., 8's) place, and the third represents the 8^0 (i.e., 1's) place, so $500_{\text{dec}} = 764_{\text{oct}} = 7 * 8^2 + 6 * 8^1 + 4 * 8^0$. - For example, one way of finding the digits is as follows: the first octal digit of 500 is $500 / 64 = 7$, the second digit is $500 \% 64 / 8 = 6$, and the last digit is $500 \% 64 \% 8 = 4$ or $500 \% 8 = 4$. Explain the use of arithmetic operators in your program.) - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Relational and Logical Operators	In this lesson, you will describe and use relational and logical	- Relational and Logical Operators - Discussion - Tutorial - Unit Activity:	- Dictionary - Scientific Calculator - Click to Speak	Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.	Lesson Activities: Self-Evaluations

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
	operators.	Introduction to Computer Programming	- Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- In this activity, you will work with relational operators in general and as Java code. Follow the directions for each part and give your answer as a Java program along with the output of your program. - Evaluate the following expressions where $a = 5$, $b = 6$, and $c = 5$. The answer should be a Boolean (i.e. true or false). $a == b$ $c \leq a$ $b != a$ $a < c$ - Write the following relational expressions in Java code. - The number of records is more than 500. - The speed is at least 55. - The age of the student is 18. - The mass of the bolt is no more than 90. - The number of cells is less than the maximum number of cells. - The length of the first string is not the same as the length of the second string. - Write a program that computes two doubles: $x1 = 0.6 - 0.2 - 0.2 - 0.2$ and $x2 = -0.3 + 0.1 + 0.1 + 0.1$. Display each number and determine the truth value of the statement "x1 is equal to zero", etc., for each double using a method	- Mastery Test - Unit 1 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				appropriate for floating point numbers. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will evaluate and write expressions in Java and construct truth tables. - Write the following expressions in Java code. - The angle is greater than 30 and less than 90 - The width is greater than 200 or the height is greater than 300. - The patron is not under 21 years of age. - The value is at least 100, but it is not more than 1000. - Write truth tables for the following expressions. The first two columns in each table should account for all of the possible true or false combinations for a and b. $a \ \&\& \ b$ $!b \ \ a$ $!a \ \ !b$	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				○ <code>!(a && !b)</code> ■ Evaluate the following expressions. ○ <code>5 > 4 && 1 > -1</code> ○ <code>23 != 99 && 4 <= 4</code> ○ <code>12 >= 10 3 == -465</code> ○ <code>22 < 0 !(1 == 1)</code> ○ <code>!(3 > 5)</code> ○ <code>!(1 != 0)</code> ○ <code>!(22 <= 22) && !(400 == 4)</code> ○ <code>!(4 > 4 && !(3 != 3))</code> ■ Simplify <code>!(p > 5 && q != 1)</code> using De Morgan's Law. The final result should only have one logical operator. Your answer should be in Java code. ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. ■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ In this activity, you will identify instances of a short circuit and explain what happens in the interpreter. You will solve various logical problems presented in Java. You will also write a Java program that uses logical operators.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- What are the values of p, q, r, and s after the following code is executed? Explain your answers. <pre>boolean p = true, q = true, r = true, s = true, t1, t2, t3, t4; t1 = true && (p = false); t2 = false && (q = false); t3 = false (r = false); t4 = true (s = false);</pre> - Evaluate the following expressions. <code>5 * 4 == 15 + 5</code> <code>3 % 4 > 2 8 % 5 <= 3</code> <code>4 / 5 * 2 > 0 && 6 * (5 + 1) > 32</code> <code>!true && !false</code> <code>false && true && true false</code> <code>true false false false</code> <code>true && true && true false</code> <code>!true true && !(false true)</code> - The prime numbers less than 10 are 2, 3, 5, and 7. Write a program that will generate a random number n from 1 to 10. The program should include logic that will determine whether the following statement is true or false: "The number n is a prime number" (the actual number n should be displayed). - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				brief evaluation of your work below. Note what you learned and what challenged you.	
Decision-Making Control Structures	In this lesson, you will describe and use decision-making statements (if, if...else, nested if).	- Decision-Making Control Structures: - Discussion - Tutorial - Unit Activity: Introduction to Computer Programming	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You learned how the if...else statement can be used to control program flow. In this activity, you will find the output of if...else statements and write a program that generates two random integers and checks some condition. - You are given the following piece of code. What will be the output? <pre> int varJ = 1; if (varJ < 2) { System.out.println("varJ is less than 2"); } else { if (varJ > 0) { System.out.println("varJ is greater than 0"); } else { if (varJ == 1) { </pre>	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 1 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> System.out.println("varJ is equal to 1"); } else { System.out.println("varJ is unknown"); } } int varN = 10, varP = 20; if (varN < varP) { if (varN - varP >= 10) { System.out.println("(varN - varP) is " + (varN - varP)); } else { System.out.println("(varP - varN) is " + (varP - varN)); } } else { if (varN > varP) { System.out.println("varP is " + varP); } else { System.out.println("varN is " + varN); </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

```
}
}
```

- Write a program that generates two random integers, one from 1 to 3 and the other from 1 to 50. The program should display the two integers and display a message indicating whether the second integer is evenly divisible by the first. (Hint: modulus division is useful for this problem).
- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.
- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.
- You learned how the **if...else if...else** statements, along with nested if statements, create complex control flow. In this activity, you will see the use of these statements in different programs.
- You are given the following pieces of code. What will be the output?

```
int varJ = 1;
if (varJ < 2)
{
```

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> System.out.println("varJ is less than 2"); } else { if (varJ > 0) { System.out.println("varJ is greater than 0"); } else { if (varJ == 1) { System.out.println("varJ is equal to 1"); } else { System.out.println("varJ is unknown"); } } } int varN = 10, varP = 20; if (varN < varP) { if (varN - varP >= 10) { System.out.println("(varN - varP) is " + (varN - varP)); } else { System.out.println("(varP - varN) is " + (varP - </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> varN)); } } else { if (varN > varP) { System.out.println("varP is " + varP); } else { System.out.println("varN is " + varN); } } </pre> - Write a program that generates a random integer from 1 to 3. If the integer is 1, the program should display a message that says "Option A is selected". If it is 2, the program should display "Option B is selected". Finally, if the integer is 3, the program should display "Option C is selected". The program should use an if...else if...else chain. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question,	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				and check your answers. At the end of the activity, write a brief evaluation of your work. ■ In this activity, you will create a program based on a problem statement. You will make use of the if...else structures you have learned. In the second part, you will research about switch and how it can be used as a substitute for multiple if...else statements. ■ Write a program that generates a random integer from 0 to 16 and writes it as a single hexadecimal digit. Use if statements to convert values of 10 to 15 to the appropriate hexadecimal symbols. Since 16 cannot be represented as a two-digit hexadecimal number, the program should write an error message if 16 is generated. Hint: Remember that hexadecimal digits have values ranging from 0 to 15, with 10–15 being represented by A–F (you will have to write a letter to represent the digit if the value of the digit is in the 10–15 range). ■ Use the Internet and other resources to learn about the switch statement. Answer the below questions about switch statements. ○ What is a switch statement used for? ○ What is its syntax? ○ Can you use a switch statement instead of multiple if...else statements?	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Loops and Algorithms	In this lesson, you will use different types of loop statements (while, for, nested) and implement algorithms.	- Loops and Algorithms: - Discussion - Tutorial - Unit Activity: Introduction to Computer Programming	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will create trace tables for while loops, following the loop by hand. You will also write a Java program that uses a while loop to find numbers matching a particular criterion. - Write trace tables for the following code. <pre>int m = 0, n = 5; while (m < n) { m++; n--; } int m = 0, n = 0, p = 0; while (n < 10) { m++; n = n + m + p; p = m + 1; }</pre>	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 1 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Using a while loop, write a program that writes all perfect cubes less than some number n (for example, use $n = 200$, which would return 1, 8, 27...). (Hint: if you use the Math.pow() method, it will store the number being cubed as a double.) - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will create loop headers that fulfill a particular task. You will also write a Java program using a for loop. - Write for loop headers that perform the following tasks: - A variable, j, increases by ones from 5 to 10. - A variable, k, increases by threes from 0 to 24. - A variable, m, decreases by ones from 10 to 1. - A variable, n, decreases by twos from 100 to 0.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				○ Write a program that counts by 20 from 0 to 400. Convert each of the numbers to 3-digit octal numbers using arithmetic operations. ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. ■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ In this activity, you will write pseudocode, convert pseudocode to Java code, and implement an algorithm in Java. ■ Write pseudocode for a program that prints a table of numbers and their squares as shown below. 1 1 2 4 3 9 10 100 ■ Convert the following pseudocode to Java: declare an integer fact as 1	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				declare an integer j as 1 repeat until fact > 100: set fact to fact * j print fact increase j by 1 - You can use the following algorithm to determine whether a positive integer n is a prime number. The value of isPrime is true if n is prime; otherwise, it is false. <pre>isPrime = true; for(int i=2; i<n; i++) { if(n%i == 0) { isPrime = false; break; } }</pre> - Use this algorithm to iterate all integers from 2 to 100 and to display those that are prime	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				numbers. How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	

Unit 2: Classes and Constructors

Unit 2:

By the end of this unit, you will be able to do the following:

- Describe, create, and properly use classes.
- Describe, create, and call methods.
- Describe and employ different approaches to using parameters in methods.

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
Classes	In this lesson, you will describe, create, and properly use classes.	■ Classes: ○ Discussion ○ Tutorial ○ Unit Activity: Classes and Constructors	■ Dictionary ■ Scientific Calculator ■ Click to Speak ■ Translate ■ Highlighter ■ Discussion Board ■ Slide Narration ■ Glossary	■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ You looked at different types of methods in Java. In this activity, you will go through a class and answer questions on packages and methods. ■ Study the following code that uses the Point class and answer the following questions. import java.awt.Point;	■ Lesson Activities: Self-Evaluations ■ Mastery Test ■ Unit 2 Post Test ■ AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> public class PointTest { public static void main(String[] args) { Point pnt=new Point(4,5); double point_x = pnt.getX(); double point_y = pnt.getY(); System.out.println("First location:"); System.out.println("X coordinate (double) " + point_x); System.out.println("Y coordinate (double) " + point_y); pnt.move(-1,-2); point_x = pnt.getX(); point_y = pnt.getY(); System.out.println("\nSecond location:"); System.out.println("X coordinate (double) " + point_x); System.out.println("Y coordinate (double) " + point_y); } } </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- What package does the Point class belong to? - Identify the statement that uses the constructor. - Identify a statement that uses an accessor method. - Identify a statement that uses a mutator method. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You learned about the concept of encapsulation. In this activity, you will answer questions related to access specifiers. You will also create a class having instance variables, a constructor, and a method based on the given information. - Write declaration statements for instance variables with the mentioned	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				characteristics: ○ a public integer representing the radius of a circle ○ a private double representing a bank account balance ○ two private integers representing starting and ending values ○ two public doubles representing wavelength and amplitude ■ Write a class Circle that represents a circle. Instance variables should include the radius, x-coordinate, and y-coordinate (all doubles). Write the code so that these variables are encapsulated. Write a simple constructor that assigns a radius of 1.0 and coordinates of (0,0) to a new circle, and write a simple method that prints the radius of the circle. ■ Write a short program that uses your Circle class. Create a circle object and use the method. ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You looked at class variables and methods. In this activity, you will add a class variable to an existing class. You will also add a static method to the class. - You created a Circle class in the previous activity. Now, expand the class to include a static final variable PI. Add another public instance variable that stores the area of the circle. Write a simple method that assigns a value to the area of the circle (hint: the area is equal to radius*radius*PI). - Write a static method for the Circle class that prints the message “The circumference of a circle is the radius times 2*pi”. - Expand the program that uses the Circle method by using the method that assigns the area and the static method. Have your	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				program print the value of the area. How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Methods	In this lesson, you will describe, create, and call methods.	- Methods: - Discussion - Tutorial - Unit Activity: Classes and Constructors	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will add a new constructor to the existing class Circle. You will then write a short Java program that uses the new constructor. - In a previous lesson, you created the class Circle. Expand this class by adding a new parameterized constructor that accepts radius as well as x and y input, and then assigns these to the instance variables. Write a program that creates a Circle object using the new constructor. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 2 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will modify a constructor in the existing class Circle. You will then write a short Java program using the new constructor. - Building upon the Circle class you created earlier, expand the class by making the default constructor increment a static variable that counts the number of Circle objects created. The default constructor should print the number of circle objects created. Modify and overload the parameterized constructor. Write a program that creates two circle objects using the overloaded constructor. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will add conditions to your existing class Circle. You will then write a short Java program that uses the new constructor. - Enforce a precondition on the parameterized constructor of the Circle class by throwing an exception. The precondition should be radius > 0 -100 <= x <= 100 -100 <= y <= 100 - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Method Parameters	In this lesson, you will describe and employ different approaches to using parameters in methods.	- Method Parameters: - Discussion - Tutorial - Unit Activity: Classes and Constructors	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will expand an existing class with new methods. You will pass	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 2 Post Test - AP Computer

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
			- Discussion Board - Slide Narration - Glossary	values to these methods. You will also use method overloading. - You have already created the class Circle in a previous lesson. Using the Circle class, write the code required to complete this activity. - Expand the Circle class by adding a method that increases the radius of the circle by an amount that is passed to the method as an argument. - Expand the Circle class further by adding two methods of the same name that return the distance from the center of the circle to a given point. One version of the method accepts no input and assumes that the coordinates of the given point are (0,0). The second version accepts the x and y coordinates of the given point. (Hint: You can use the distance formula to calculate the distance.) - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each	Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will pass objects to a method. You will also pass values and get values returned. Using the previous Circle class, write the code required to complete the activity. - Expand the Circle class further by adding a static method that accepts one circle as an argument and returns the distance from its center to the origin. - Expand the Circle class further by adding a static method that accepts two circles as arguments and returns the distance between their centers. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- In this activity, you will expand your Circle class by adding new methods. You will correctly use the this() keyword. You will also pass objects and references to a method. Using the previous Circle class, write the code required to complete this activity. - Expand the Circle class further by adding an instance method that changes the x and y coordinates to new values passed as arguments. The code should use this to make it clear that the instance variables for the x and y coordinates are instance variables. - Expand the Circle class again by adding an instance method that passes the reference to the Circle object to the existing static method, which then returns the distance from the center to the origin. The new method should return the value of the distance returned by the existing method. The existing method is given below. <pre>public static double distOrigin(Circle c) { double a = 0, b = 0; double r = Math.sqrt(Math.pow((c.x_center-a),2) + Math.pow((c.y_center-b),2));</pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre>return r; }</pre> How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	

Unit 3: Data Structures, Searching and Sorting Algorithms

Unit 3:

By the end of this unit, you will be able to do the following:

- Create String objects and use various String methods.
- Create and manipulate one-dimensional and two-dimensional arrays.
- Create an ArrayList object and use various ArrayList methods.
- Perform selection and insertion sort on an array and compare their performance.
- Understand the concept of recursion and implement recursive methods to solve problems.
- Perform sequential and binary search on an array and compare their performance.

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
The String Class	In this lesson, you will create String objects and use various String methods.	■ The String Class: ○ Discussion ○ Tutorial ○ Unit Activity: Data Structures, Searching and Sorting Algorithms	■ Dictionary ■ Scientific Calculator ■ Click to Speak ■ Translate ■ Highlighter ■ Discussion Board ■ Slide Narration ■ Glossary	■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ In this activity, you will create String objects and see the difference between using a string pool and a new keyword. ■ Write statements to create new String objects using the string constructor for the following situations.	■ Lesson Activities: Self-Evaluations ■ Mastery Test ■ Unit 3 Post Test ■ AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Create a String object called stringOne with a value of "Hello, Mr. Knuth" (without using the new keyword). - Create a String object called lang with a value of "Java" (using the new keyword). - Write the output of the following code. Explain why the code produces that particular output. <pre>String s1 = "abc"; String s2 = "abc"; String s3 = new String("abc"); System.out.println(s1 == s2); System.out.println(s2 == s3); s2 = s3; System.out.println(s2 == s3);</pre> - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will find the output of Java statements that use string concatenation and substrings. You will also write methods that accept a String object. - What is the output of the given Java code in the following examples? <pre>String string2 = "Java"; String string4 = new String("mathematics"); String string5 = new String("data"); System.out.println(string2 + " is useful for " + string4 + " and " + string5 + " analysis."); String st1 = "Happy"; String st2 = "birthday!"; System.out.println(st1 + st2 + "You are " + 10 + 8 + " years old!"); String st3 = "Learn Java!"; System.out.println(st3.substring(1,4)+st3.subst ring(10,11));</pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				String st4 = "sourcecode"; System.out.println(st4.substring(6)); - Write a static method called "inator" that accepts a string object as input, appends "- inator" to it, and returns the resulting string. - Write a static method that accepts a String object as input and returns the first four characters of the string. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You have learned about different ways in which you can input data and manipulate strings. You will use that knowledge in this activity to perform various tasks. - Write a static method that accepts a 9-digit taxpayer identification number as a string (such as 132549876) from the console, and	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				then returns the string with the format 132-54-9876. - Write a static method that accepts a String object as input and returns the last four characters of the string. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Arrays	In this lesson, you will create and manipulate one-dimensional and two-dimensional arrays.	- Arrays - Tutorial - Unit Activity: Data Structures, Searching and Sorting Algorithms	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You have learned about one-dimensional arrays. In this activity, you will answer questions based on the declaration and initialization of one-dimensional arrays. You will also perform basic computations on these arrays. - Write statements that declare the following arrays. - Declare an array called force that contains 15 doubles.	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 3 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				○ Declare an array called level that contains eight integers. ○ Write statements that declare and initialize the following arrays. ■ Declare and initialize an array called weight that contains the values 4.6, 7.5, 8.1, 9.5, and 0.2. ■ Declare and initialize an array called inventory that contains the values 5, 6, 2, 8, and 22. ■ Write statements that perform the following tasks. ○ Print the 3rd element of the array called inventory. ○ Add 11.5 to the 1st element of the array called weight. ○ Calculate the sum of the 5th and 4th elements of the array called weight, and assign it to the variable xWeight. ○ Set the value of the 3rd element of the array called inventory equal to the value of the 2nd element. ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You learned about multidimensional arrays and how to use loops to access these arrays. In this activity, you will write short programs and answer questions related to multidimensional arrays. - Write a short program that declares and initializes an array of four doubles and prints each value. Use a loop to iterate over the elements. - Write statements that perform the following tasks. - Declare a two-dimensional array of doubles, named <code>bondLengths</code>, with 30 rows and 10 columns. - Declare a two-dimensional array of integers with the following values: 0 5 4 6 9 1	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				○ Write a short program that creates a 3-by-4 array of integers and populates each element with the product of its row number times its column number. ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. ■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ You have learned how to use arrays to store objects and their use in methods. In this activity, you will write short programs in which you will create arrays. ■ Write a short program using the Circle class that you learned about in previous lessons. Write code that creates an array of 50 circles and populates it with circle objects. ■ Write a short program that includes a static method that accepts an array of strings as a	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				parameter and converts each element to all lowercase (Hint: Use the <code>String.toLowerCase()</code> method). The method should have no return value. Include code that invokes and tests the method. - Using the short program you developed in Question 2, add a static method that accepts an array of strings and returns a parallel array containing the length of each string. Write additional code that invokes and tests the method. (Hint: Use the <code>String.length()</code> method to find the length of each string. Remember that string length and array length are two different things.) - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
The ArrayList Class	In this lesson, you will create an ArrayList object and use various ArrayList methods.	- The ArrayList Class: - Discussion - Tutorial - Unit Activity: Data Structures, Searching and	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You looked at the features of an array list and its usage. In this activity, you will create	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 3 Post Test - AP Computer Science A End-

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
		Sorting Algorithms	- Discussion Board - Slide Narration - Glossary	array lists based on the statements given. You will also rewrite code containing an array as an array list. - Write statements to create new array lists given the following parameters. - An array list called patron that contains strings. - An array list called station that contains doubles. - An array list called score that contains integers. - Rewrite the following code so that it uses an array list instead of an array: <pre>int[] numerals = {4,5,6,7}; numerals[0] = 3; System.out.println(numerals[3]);</pre> - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.	of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- You looked at the enhanced for loop and its benefits. In this activity, you will come across programs that use the conventional for loop. You will then rewrite these programs using the enhanced for loop. - Rewrite the following conventional for loops as enhanced for loops. <pre>double sum = 0; for (inti = 0; i<dub.size(); i++) { sum = sum + dub.get(i); } System.out.println(sum); int[] values = {5,9,2,8}; int largest = 0; for (inti = 1; i<values.length; i++) { if (values[i] > largest) largest = values[i]; } System.out.println(largest);</pre> - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				of your work. - You looked at the various operations performed on array lists. In this activity, you will manipulate array lists. You will also write a method that accepts an array list. - Write statements that do the following. - Insert an element containing "04/15" into the third position in an array list called <code>plantingDate</code>. - Delete the first element from an array list called <code>inventory</code>. - Delete an element containing "A53" from an array list called <code>carID</code>. - Write the contents of an array list called items after the following operations have been completed. <pre> ArrayList<Integer> items = new ArrayList<Integer>(); items.add(6); items.add(2); items.add(1); items.add(3); items.remove(0); items.add(1,5); items.remove(2); </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Write a static method that accepts an array list containing integers. The method should print the contents of all elements that store even-numbered integers. The method should not return anything. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Sorting Algorithms	In this lesson, you will perform selection and insertion sort on an array and compare their performance.	- Sorting Algorithms: - Discussion - Tutorial - Unit Activity: Data Structures, Searching and Sorting Algorithms	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will create a class that uses the selection sort algorithm to sort integers. You will also create a class to use selection sort to sort doubles. - Complete the example class below to create a fully functional class that uses selection sort to sort a given array. A separate class should be used to create an array and run the sort method. <pre>public class SelectionSort {</pre>	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 3 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> private static int positionMin(int[] vals, int startPosition) { // Finds the element between startPosition and the end of the array // that has the lowest value int minPosition = startPosition; // minPosition will hold the index // of the element with the lowest value for (int i = startPosition; i < vals.length; i++) { if (vals[i] < vals[minPosition]) { minPosition = // Complete this statement! ##### } } return minPosition; } private static void swap(int[] vals, int firstPosition, int secondPosition) { // Swaps the values of the elements at firstPosition and secondPosition int temp; // temporary variable that holds the value of the element at firstPosition temp = vals[firstPosition]; vals[firstPosition] = // Complete this statement! ##### vals[secondPosition] = // Complete this statement! ##### </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

```

return;
}

public static void selSort(int[] vals) {
    int minPos;
    for (int startPos = 0; startPos < vals.length;
startPos++) {
        // startPos is the starting index of the
array for each iteration of the selection
        // minPos is the index of the selected
element
        minPos = positionMin(vals, // Complete
this statement! #####
        swap(vals,startPos, // Complete this
statement! #####
    }
    return;
}
}

```

- Copy your completed code and modify it so that it will sort an array of doubles.
- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity you will create a class that uses the insertion sort algorithm to sort integers. You will also create a class to use insertion sort to sort doubles. - Complete the example code below to create a fully functional class that uses insertion sort to sort a given array. A separate class should be used to create an array and run the sort method. <pre> public class InsertionSort { private static void insertElem(int[] vals, int bottomPos) { int temp; // stores the value to be inserted temp = vals[bottomPos]; for (int i = bottomPos - 1; i >= 0; i--) { if (temp >= vals[i]) { vals[i+1] = // Complete this statement! ##### return; } vals[i+1] = // Complete this statement! </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

```
#####
    }
    vals[0] = // Complete this statement!
#####
    return;
}

public static void insSort(int[] vals) {
    for (int i = 1; i < vals.length; i++) {
        insertElem(vals, // Complete this
statement! #####
    }
    return;
}
}
```

- Copy your completed code and modify it so that it will sort an array of doubles.
- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.
- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- In this activity you will create an array of 1000 random doubles and sort them using either selection sort or insertion sort. You will then create a very large array and sort them with the two different algorithms, measuring the time for each using a simple stopwatch class. Then you will determine which algorithm is faster. - Modify the code from section 1 to create and sort a large array (1,000 elements) full of random doubles between 0 and 1000. Show the modified code. - Modify the code from sections 1 and 2 to create and sort a large array (100,000 elements) full of random doubles between 0 and 1000. Use a stopwatch class to measure the execution time of each of the two algorithms. (Hint: measure just the execution times of the sort algorithms, not the time to generate the random numbers. Do not print these large arrays!) Compare the execution times of the two algorithms. Which algorithm was faster and by how much? - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				write a brief evaluation of your work below. Note what you learned and what challenged you.	
Recursion and Merge Sort	In this lesson, you will understand the concept of recursion and implement recursive methods to solve problems.	- Recursion and Merge Sort: - Discussion - Tutorial - Unit Activity: Data Structures, Searching and Sorting Algorithms	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will identify a base case in an existing recursive method. You will also write a recursive method that computes powers of two. - Study the following recursive method and identify the base case. <pre>public static void upCounter(int n) { if (n == 1) { System.out.println(n); return; } else { upCounter(n-1); System.out.println(n); return; } }</pre>	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 3 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- Write a recursive method to compute 2^n, where n is a positive integer. (Hint: Approach the problem in a recursive manner, such as $2^1 = 2$, $2^2 = 2 * 2^1$, $2^3 = 2 * 2^2$, etc.) - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will write a recursive method to find Fibonacci numbers. You will also rewrite the method to include a recursive helper method to produce an error with invalid user input. - The Fibonacci sequence is defined by seed values $F_1 = 1$ and $F_2 = 1$, followed by $F_n = F_{n-1} + F_{n-2}$, resulting in the sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, 55... The numbers in this sequence are known as the Fibonacci	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				numbers. An iterative method for calculating the nth Fibonacci number is as follows: <pre>public static int fibonacci_iterative(int n) { int[] fib = new int[n]; fib[0] = 1; fib[1] = 1; for (int i = 2; i < n; i++) { fib[i] = fib[i-1] + fib[i-2]; } return fib[n-1]; }</pre> - Write a recursive method for computing Fibonacci numbers. Hint: Two base cases are necessary—one for F1, and one for F2. No array is necessary for the recursive method. Compare the relative performance of the two methods in calculating the 45th Fibonacci number. - Convert the recursive Fibonacci method to a pair of methods that implements the recursive part of the problem as a private helper method. The public method should enforce the precondition that the input is a positive integer.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will complete a class that performs merge sort. You will also use a SimpleStopwatch class to compare the performance of merge sort with other sorting algorithms. - Given the partially-written code below, complete the implementation of merge sort for an array of doubles. <pre> public class MergeSortD { public static void mergeSort(double[] input) { // Test for the base case if // Complete this statement! ##### </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> // Divide the array in half // determine the lengths of the two new arrays int length1 = // Complete this statement! ##### int length2 = // Complete this statement! ##### // create the two new arrays double[] array1 = new double[// Complete this statement! ##### double[] array2 = new double[// Complete this statement! ##### // copy elements to the first new array from the original array for (int i = 0; i < length1; i++) { array1[i] = input[i]; } // copy elements to the second new array from the original array for (int i = 0; i < length2; i++) { array2[i] = input[i + // Complete this statement! ##### } // Call mergeSort for the first new array mergeSort(// Complete this statement! ##### // Call mergeSort for the second new array mergeSort(// Complete this statement! ##### </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> // Merge the two new arrays back into the original array // Select elements from the two new arrays until all elements // of one have been selected int front1 = 0, front2 = 0, inputPosition = 0; while (front1 < length1 && front2 < // Complete this statement! ##### if (array1[front1] < array2[front2]) { input[inputPosition] = array1[// Complete this statement! ##### front1++; } else { input[inputPosition] = // Complete this statement! ##### front2++; } inputPosition++; } // Copy all remaining elements to the original array if (front1 < // Complete this statement! ##### for (int i = front1; i < length1; i++) { input[inputPosition] = array1[// Complete this statement! ##### inputPosition++; } } else if (front2 < length2) { </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> for (int i = front2; // Complete this statement! ##### input[inputPosition] = array2[// Complete this statement! ##### inputPosition++; } } // Now the original is sorted! return; } } </pre> Compare the performance of merge sort to selection and insertion sort using a Stopwatch class. The SimpleStopwatch class is given below. <pre> public class SimpleStopwatch { // These time values need to be stored using longs private long startingTime, endingTime; public SimpleStopwatch() { // initialize the instance variables startingTime = System.currentTimeMillis(); endingTime = System.currentTimeMillis(); } } </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> public void start() { //record the starting time startingTime = System.currentTimeMillis(); return; } public void stop() { //record the ending time endingTime = System.currentTimeMillis(); return; } public long getDuration() { // report the time difference return endingTime - startingTime; } </pre> How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Searching Algorithms	In this lesson, you will perform sequential and binary search on an array and	- Searching Algorithms: - Discussion - Tutorial - Unit Activity:	- Dictionary - Scientific Calculator - Click to Speak	Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 3 Post Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
	compare their performance.	Data Structures, Searching and Sorting Algorithms	- Translate - Highlighter - Discussion Board - Slide Narration - Glossary	of your work. - You looked at the pseudocode to perform sequential search. In this activity, you will implement sequential search using an array of integers. - Write a static method that implements sequential search for an array of integers. The method should accept an array and an integer to search for and return the position of the match. Write another method that uses the search method to find the position of a value entered from the console. Test the method using an array of unsorted integers. The code should be able to interpret a return value of -1 as meaning a failed search. - Hint: use the Scanner nextInt() method to enter an integer from the console. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation	AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

of your work.

- You looked at how binary search works. In this activity, you will implement binary search using an array of integers.
- Write a method that implements binary search for an array of integers. The method should accept an array and an integer to search for and return the position of the match. Write another method that uses the search method to find the position of a value entered from the console. To test the method, use a sorted array of integers with values ranging from 11 to 20. The code should be able to interpret a return value of -1 as meaning a failed search.
- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.
- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.
- You saw various searching algorithms so far. In this activity, you will use the

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				permutations method of find the permutations of a word. You will also modify the sequential and binary search algorithms to handle array lists. - Write a program that uses the permutations method to print the permutations of the word "sine." The program should also report the number of permutations. Include the class Permutations you saw earlier. - Modify your sequential search method to handle array lists of strings. Use the permutations method to compile an array list of the permutations of "adeilmnopr," and use your modified sequential search method to find the position of "palindrome" in the list. The list of permutations should not be printed. Use a stopwatch class to measure the time required to complete the same search 1,000 times (a search through the 3628800 elements by itself takes such a small amount of time that it is difficult to measure). Hint: use the String equals() method to compare the key to the permutation. - Modify your binary search method to handle array lists of strings. Use your modified code to find the position of	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				“palindrome” in the permutations of “adeilmnopr” (if you use the permutations method provided in this lesson, placing the letters in alphabetical order like this will produce a sorted list of permutations). Use a stopwatch class to measure the time required to complete the same search 1,000 times. Compare the performance of the two approaches to searching. ■ Hint: use the String compareTo() method to compare strings. ■ How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	

Unit 4: Object-Oriented Java and Ethics in Computing

Unit 4:

By the end of this unit, you will be able to do the following:

- Explain and implement object-oriented programming design.
- Explain the advantages of inheritance and polymorphism, and implement inheritance and polymorphism.
- Describe and implement abstraction, and create an interface.
- Describe privacy and legality in the context of computing.

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
Object-Oriented Programming Design	In this lesson, you will explain and implement object-oriented programming design.	■ Object-Oriented Programming Design: ○ Discussion ○ Tutorial ○ Unit Activity: Object-Oriented Java and Ethics in Computing	■ Dictionary ■ Scientific Calculator ■ Click to Speak ■ Translate ■ Highlighter ■ Discussion Board ■ Slide Narration ■ Glossary	■ Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. ■ In this activity, you will identify classes in a problem domain. You will then describe the class and give your reasoning. ■ Identify classes in the following problem domain. For each class that you choose, describe why it would be appropriate and what it would represent.	■ Lesson Activities: Self-Evaluations ■ Mastery Test ■ Unit 4 Post Test ■ AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				"The classic shell game is implemented as follows: the player pays \$1 to play. The computer places a ball under one of three identical cups. The computer then shuffles the cups around and the player then guesses which cup the ball is under. If the player guesses correctly, then the player wins \$2. The player's balance can be applied to additional games." - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will identify classes for a particular program. You will also identify the responsibilities of the classes and the collaborators of each class. You will then determine the relationship of sample classes that are given to you. - Consider a program that creates a catalog of courses offered by a college for a given semester. The list should include the course	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				name, meeting time, instructor's name, instructor's email address, room number, building name, and the number of seats in the room. Identify some appropriate Java classes, the behaviors and responsibilities of the classes, and the collaborators of each class. Determine whether each pair has an is-a or a has-a relationship. Indicate any pair that likely involves inheritance or aggregation. - Company, Employee - Hawk, Bird - House, Room - House, Building - Laptop, Computer - Hospital, Patient How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. In this activity, you will write tester classes for both unit testing and boundary testing.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				Write a tester class for unit testing the code below. <pre> public class SquareShape { public double side, area, perimeter; public SquareShape() { area = 0.0; perimeter = 0.0; side = 0.0; } public SquareShape(double s) { side = s; this.calcArea(); this.calcPerimeter(); } public double getArea() { return area; } public double getPerimeter() { return perimeter; } </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> } public void calcArea() { area = side*side; return; } public void calcPerimeter() { perimeter = 4*side; return; } } </pre> Write a tester class that tests two boundary cases for the code below. <pre> public class Boundary { public static double WeeklyEarnings(double hours, double rate) { double earnings = 0; if (hours < 0 rate <= 0) { System.out.println("Invalid input"); } else if (hours > 40.0) { earnings = 40 * rate + (hours - 40) * 1.5 * rate; </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> } else { earnings = 40 * rate; } return earnings; } } </pre> How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Inheritance and Polymorphism	In this lesson, you will explain the advantages of inheritance and polymorphism, and implement inheritance and polymorphism.	- Inheritance and Polymorphism: - Discussion - Tutorial - Unit Activity: Object-Oriented Java and Ethics in Computing	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration - Glossary	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - In this activity, you will classify concepts into a hierarchical structure and create a chart to examine the class hierarchy of the types of people in a new college database. You will also write a superclass and extend this class with a subclass. - Classify the following concepts into a hierarchical structure: dog, mammal, animal, cat, reptile	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 4 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

- Examine the class hierarchy of the types of people in a new college database: person, employee, faculty, student, and staff. Determine the appropriate hierarchy and suitable class names, then construct a chart to illustrate the hierarchy. Hint: You can use **BlueJ** to illustrate a class hierarchy by writing simple classes with no contents, such as **public class TestClass{}** and **public class NextClass extends TestClass{}**.
- Write a superclass **Person** and extend it using a **Student** subclass. The **Person** class should include instance variables for email address and name, an appropriate default constructor and one where the instance variables can be specified, and accessor and mutator methods for each of the instance variables. The **Student** class should include year (a string), major (a string), and grade point average (a double), with appropriate mutator and accessor methods as well as a default constructor that uses the **super()** statement.
- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

what you learned and what challenged you.

- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.
- In this activity, you will extend a class with another class, creating a super- and subclass relationship. You will also override some of the superclass's methods.
- Write a subclass **Triangle** that extends the **Shape** class given below. The new class should do the following:
 - Include instance variables base and height.
 - Include a constructor that passes the number of sides "3" to the **Shape** constructor.
 - Override the **calcArea()** method so it calculates the area as **$0.5 * \text{base} * \text{height}$** .
 - Override the **showData()** method so that it also prints the number of sides, base, and height.

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

```

public class Shape
{

    private int numSides;
    public double area;

    public Shape(){
        area = 0;
        numSides = 0;
    }

    public Shape(int n)
    {
        area = 0;
        numSides = n;
    }

    public double calcArea()
    {
        return area;
    }

    public void showData()
    {
        System.out.println("The number of sides is "
        + numSides);
        return;
    }
}

```

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

}
}

- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.
- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.
- In this activity, you will modify methods of an existing class and write a new subclass that exhibits polymorphism. You will also write a driver class that tests the polymorphism of your new classes.
- Modify the **Person** method to include a method called **printData()** that prints the name and email address of the person. Modify the **Student** class so that the **printData()** method also prints the year, major, and grade average. Write a new subclass of **Person** called **Employee** that includes a salary field and

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				a printData() method that also prints salary in addition to the other data. Write your code so that it reuses existing code. - Write a driver class that includes a static method that accepts an object of the Person, Student, or Employee classes as a parameter and calls the printData() method. Test the method by passing Person, Student, or Employee objects to it as arguments. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Abstraction and Interfaces	In this lesson, you will describe and implement abstraction, and create an interface.	- Abstraction and Interfaces: - Discussion - Tutorial - Unit - Activity: Object-Oriented Java and Ethics in Computing	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You learned about abstract classes and implementing their methods in a subclass. In this activity, you will rewrite an existing class as an abstract class. You will also extend the new class and add new methods.	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 4 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
			Glossary	Convert the Person class to an abstract class. Change printData() to an abstract method. <pre> public class Person { public String emailAddress; public String name; public Person() { name = "Unknown"; emailAddress = "Unknown"; } public Person(String n, String e) { name = n; emailAddress = e; } public String getName() { return name; } public String getEmail() { return emailAddress; } public void setName(String n) { name = n; return; } </pre>	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				<pre> } public void setEmail(String e) { emailAddress = e; return; } public void printData() { System.out.println("Name: " + name); System.out.println("Email address: " + name); } } </pre> - Implement the Employee class as an extension of Person. Each employee will have a salary value. The Employee class should include appropriate constructors and methods. The method printData() in Employee should print name, email address, and salary. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				- You learned about interfaces and how they are different from abstract classes. In this activity, you will create an interface. You will also create a class that implements the interface. - Write an interface called Shape that includes a method calcArea() that should return the calculated area and a method showData() that should print the geometric parameters of the shape. - Implement the Shape interface in a class called Rectangle. The class should store length and width as instance variables and include appropriate constructors. The showData() method should print length, width, and area. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - You learned about the Comparable interface and its use to compare two items. In this	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
				activity, you will implement the Comparable interface in the Rectangle class. You will then use the minimum method to find the smaller of two rectangles. - Implement the Comparable interface in the Rectangle class written in the previous activity. Base the comparison on the area of the rectangle. - Use the minimum method used as an example in this section to find the smaller of two rectangles. - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.	
Ethics in Computing	In this lesson, you will describe privacy and legality in the context of computing.	- Ethics in Computing: - Discussion - Tutorial - Unit Activity: Object-Oriented Java and Ethics in Computing	- Dictionary - Scientific Calculator - Click to Speak - Translate - Highlighter - Discussion Board - Slide Narration	- Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - Discuss one way in which you could be seriously affected by a failure related to computer reliability or security.	- Lesson Activities: Self-Evaluations - Mastery Test - Unit 4 Post Test - AP Computer Science A End-of-Semester Test

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
			Glossary	- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work. - Discuss your current use of computers, smart phones and tablets. What is your comfort level with the safety and security of your personal data online? How much privacy are you willing to give up in order to enjoy the rewards of online technology? - How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you. - Read the instructions for this self-checked activity. Type in your response to each question, and check your answers. At the end of the activity, write a brief evaluation of your work.	

Lesson	Lesson Level Objective	Instructional Material	Tools	Activity	Assessment
--------	------------------------	------------------------	-------	----------	------------

Although not necessary, placing a copyright notice such as

Copyright (C) 2001 Jane Doe - All Rights Reserved

in a commented line in source code (usually near the top) can make it clear that the work is under copyright and not in the public domain. Write a short program that includes a copyright statement.

- Imagine that you are a programmer in the future who will publish a piece of software either as an individual or as part of a business or other organization. Describe the purpose of the software and your goals and hopes for how people will use the software. Describe how you would use copyright to help achieve these goals.
- How did you do? Rate your work on a scale of 1 to 5, with 5 as the highest score. Then write a brief evaluation of your work below. Note what you learned and what challenged you.