**Chapter 2**

**Self-Review Exercises**

**2.1 Fill in the blanks in each of the following statements:**

a) A(n)\_\_\_\_left brace, {\_\_\_\_ begins the body of every method, and a(n)\_\_\_\_right brace, }\_\_\_\_ ends the body of every method.

b) You can use the \_\_\_\_if selection\_\_\_\_\_ statement to make decisions.

c) \_\_\_\_//\_\_\_\_\_begins an end-of-line comment.

d)\_\_\_\_Blank spaces\_\_\_\_ , \_\_\_\_tabs\_\_\_\_and \_\_\_\_newlines\_\_\_\_are called white space.

e) \_\_\_\_\_Keywords\_\_\_\_ are reserved for use by Java.

f) Java applications begin execution at method \_\_\_\_main\_\_\_\_ .

g) Methods\_\_\_\_System.out.print\_\_\_\_ ,\_\_\_System.out.println\_\_\_\_ and\_\_\_System.out.printf\_\_\_ display information in a command window.

**2.2 State whether each of the following is true or false. If false, explain why.**

a) Comments cause the computer to print the text after the // on the screen when the program executes.

False – Comments are ignored by the compiler; they do **not** cause any output on the screen.

b) All variables must be given a type when they’re declared.

**True** – In Java, every variable **must** have a data type when it is declared

c) Java considers the variables number and NuMbEr to be identical.

**False** – Java is **case-sensitive**, so number and NuMbEr are **different variables**

d) The remainder operator (%) can be used only with integer operands.

**False** – The remainder operator % can be used with **both integer and floating-point operands**.

e) The arithmetic operators \*, /, %, + and - all have the same level of precedence.

**False** – The operators do **not** all have the same precedence. \*, /, and % have **higher precedence** than + and -.

**2.3 Write statements to accomplish each of the following tasks:**

a) Declare variables c, thisIsAVariable, q76354 and number to be of type int.

b) Prompt the user to enter an integer.

c) Input an integer and assign the result to int variable value. Assume Scanner variable input can be used to read a value from the keyboard.

d) Print "This is a Java program" on one line in the command window. Use method System.out.println.

e) Print "This is a Java program" on two lines in the command window. The first line should end with Java. Use method System.out.printf and two %s format specifiers.

f) If the variable number is not equal to 7, display "The variable number is not equal to 7”

**2.4 Identify and correct the errors in each of the following statements:**

a) if (c < 7)

System.out.println("c is less than 7");

b) if (c >= 7)

System.out.println("c is equal to or greater than 7");

**2.5 Write declarations, statements or comments that accomplish each of the following tasks:**

a) State that a program will calculate the product of three integers.

b) Create a Scanner called input that reads values from the standard input. c) Declare the variables x, y, z and result to be of type int.

d) Prompt the user to enter the first integer.

e) Read the first integer from the user and store it in the variable x.

f) Prompt the user to enter the second integer.

g) Read the second integer from the user and store it in the variable y.

h) Prompt the user to enter the third integer.

i) Read the third integer from the user and store it in the variable z.

j) Compute the product of the three integers contained in variables x, y and z, and assign the result to the variable result.

k) Use System.out.printf to display the message "Product is" followed by the value of the variable result.

**2.6 Using the statements you wrote in Exercise 2.5, write a complete program that calculates and prints the product of three integers.**

**Exercises**

**2.7 Fill in the blanks in each of the following statements:**

a) \_\_\_\_Comments\_\_\_ are used to document a program and improve its readability.

b) A decision can be made in a Java program with a(n)\_\_\_\_if\_\_\_\_\_ .

c) Calculations are normally performed by \_\_\_arithmetic\_\_\_\_ statements.

d) The arithmetic operators with the same precedence as multiplication are \_\_\_\_division\_\_\_\_ and \_\_\_\_modulus\_\_\_\_\_ .

e) When parentheses in an arithmetic expression are nested, the \_\_\_innermost\_\_\_\_ set of parentheses is evaluated first.

f) A location in the computer’s memory that may contain different values at various times throughout the execution of a program is called a(n) \_\_variable\_\_.

**2.8 Write Java statements that accomplish each of the following tasks**: a) Display the message "Enter an integer: ", leaving the cursor on the same line.

b) Assign the product of variables b and c to variable a.

c) Use a comment to state that a program performs a sample payroll calculation.

**2.9 State whether each of the following is true or false. If false, explain why.**

a) Java operators are evaluated from left to right.

False – Java operators are not all evaluated from left to right, evaluation depends on operator precedence and associativity

b) The following are all valid variable names: \_under\_bar\_, m928134, t5, j7, her\_sales$, his\_$account\_total, a, b$, c, z and z2.

True – All of these are valid Java variable names:  
They follow Java’s naming rules:

* Must start with a letter, underscore (\_), or dollar sign ($).
* Can include digits, but not as the first character.

c) A valid Java arithmetic expression with no parentheses is evaluated from left to right.

False – A valid Java arithmetic expression with no parentheses is not always evaluated from left to right, the order depends on operator precedence.  
For example, in 3 + 4 \* 5, multiplication is performed before addition.

d) The following are all invalid variable names: 3g, 87, 67h2, h22 and 2h.

False – Not all are invalid.

3g, 87, and 2h are invalid (they start with digits).

But 67h2 and h22 are valid variable names (they start with letters).

**2.10 Assuming that x=2 and y=3, what does each of the following statements display?**

a) System.out.printf("x = %d%n", x);

x = 2

b) System.out.printf("Value of %d + %d is %d%n", x, x, (x + x));

Value of 2 + 2 is 4

c) System.out.printf("x =");

x =

d) System.out.printf("%d = %d%n", (x + y), (y + x));

5 = 5

2.11 Which of the following Java statements contain variables whose values are modified?

a) p=i+j+k+ 7;

Yes, this statement assigns a new value to `p`.

b) System.out.println("variables whose values are modified");

No, this only displays text; no variables are changed.

c) System.out.println("a = 5");

No, it just prints the string "a = 5", doesn’t assign 5 to a.

d) value = input.nextInt();

Yes, it modifies the variable value by assigning user input.

**2.12 Given that y = ax3 + 7, which of the following are correct Java statements for this equation?**

a) y=a\*x\*x\*x+ 7;

Correct — this directly represents ( y = a \times x^3 + 7 ).

b) y = a \* x \* x \* (x + 7);

Incorrect — this represents ( y = a x^2 (x + 7) = a x^3 + 7a x^2 ).

c) y = (a \* x) \* x \* (x + 7);

Incorrect — same as above, it gives ( y = a x^2 (x + 7) ).

d) y = (a \* x) \* x \* x + 7;

Correct — equivalent to ( y = a x^3 + 7 ).

e) y = a \* (x \* x \* x) + 7;

Correct — also equivalent to ( y = a x^3 + 7 ).

f) y = a \* x \* (x \* x + 7);

Incorrect — expands to ( y = a x^3 + 7a x ).

**2.13 State the order of evaluation of the operators in each of the following Java statements, and show the value of x after each statement is performed:**

a) x = 7 + 3 \* 6 / 2 - 1;

= 7 + (3 \* 6) / 2 - 1

= 7 + 18 / 2 - 1

= 7 + 9 - 1

= 16 - 1

X = 15

b) x = 2 % 2 + 2 \* 2 - 2 / 2;

= (2 % 2) + (2 \* 2) - (2 / 2)

= 0 + 4 - 1

= 4 - 1

X = 3

c) x=(3 \* 9 \* (3 + (9 \* 3 / (3))));

= (3 \* 9 \* (3 + (9 \* 3 / 3)))

= (3 \* 9 \* (3 + (27 / 3)))

= (3 \* 9 \* (3 + 9))

= (3 \* 9 \* 12)

= (27 \* 12)

X = 324

**2.14 Write an application that displays the numbers 1 to 4 on the same line, with each pair of adjacent numbers separated by one space. Use the following techniques:**

a) Use one System.out.println statement.

b) Use four System.out.print statements.

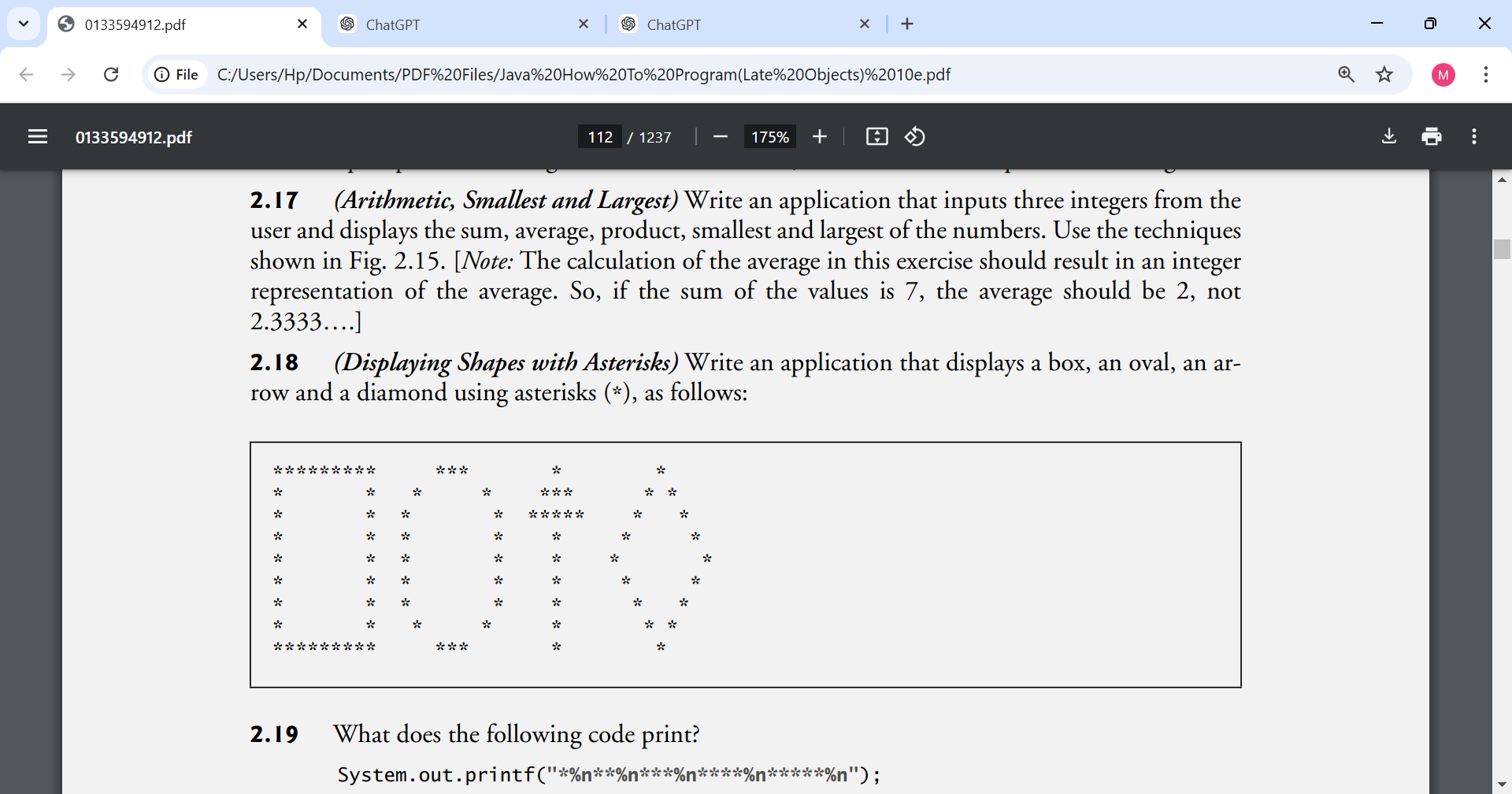
c) Use one System.out.printf statement. www.EBooksWorld.ir 68 Chapter 2 Introduction to Java Applications; Input/Output and Operators

2.15 (Arithmetic) Write an application that asks the user to enter two integers, obtains them from the user and prints their sum, product, difference and quotient (division). Use the techniques shown in Fig. 2.7.

2.16 (Comparing Integers) Write an application that asks the user to enter two integers, obtains them from the user and displays the larger number followed by the words "is larger". If the numbers are equal, print the message "These numbers are equal". Use the techniques shown in Fig.

2.15. 2.17 (Arithmetic, Smallest and Largest) Write an application that inputs three integers from the user and displays the sum, average, product, smallest and largest of the numbers. Use the techniques shown in Fig. 2.15. [Note: The calculation of the average in this exercise should result in an integer representation of the average. So, if the sum of the values is 7, the average should be 2, not 2.3333….]

2.18 (Displaying Shapes with Asterisks) Write an application that displays a box, an oval, an arrow and a diamond using asterisks (\*), as follows:



2.19 What does the following code print? System.out.printf("\*%n\*\*%n\*\*\*%n\*\*\*\*%n\*\*\*\*\*%n");

%n means a **newline** in printf.  
So each group of \* is printed on a new line.

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

2.20 What does the following code print?

System.out.println("\*"); System.out.println("\*\*\*"); System.out.println("\*\*\*\*\*");

System.out.println("\*\*\*\*");

System.out.println("\*\*");

Each println prints the string and moves to a new line.

**Output:**

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*

2.21 What does the following code print?

System.out.print("\*"); System.out.print("\*\*\*");

System.out.print("\*\*\*\*\*");

System.out.print("\*\*\*\*");

System.out.println("\*\*");

System.out.print() prints text **without moving to a new line**.

System.out.println() prints text **and then moves to a new line**.

So everything before the last line is printed **on the same line**, and the final println just adds a newline at the end.

**Output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.22 What does the following code print?

System.out.print("\*");

prints \* (no newline).

System.out.println("\*\*\*");

prints \*\*\* on the same line (because the previous didn’t end with a newline).

System.out.println("\*\*\*\*\*");

prints \*\*\*\*\* on a new line, then moves to another new line.

System.out.print("\*\*\*\*");

prints \*\*\*\* (no newline).

System.out.println("\*\*");

prints \*\* on the same line as the previous, then moves to a new line.

2.23 What does the following code print? System.out.printf("%s%n%s%n%s%n", "\*", "\*\*\*", "\*\*\*\*\*");

%s prints a string.

%n prints a newline (moves to the next line).

The three format specifiers %s%n%s%n%s%n correspond to the three string arguments "\*", "\*\*\*", and "\*\*\*\*\*".

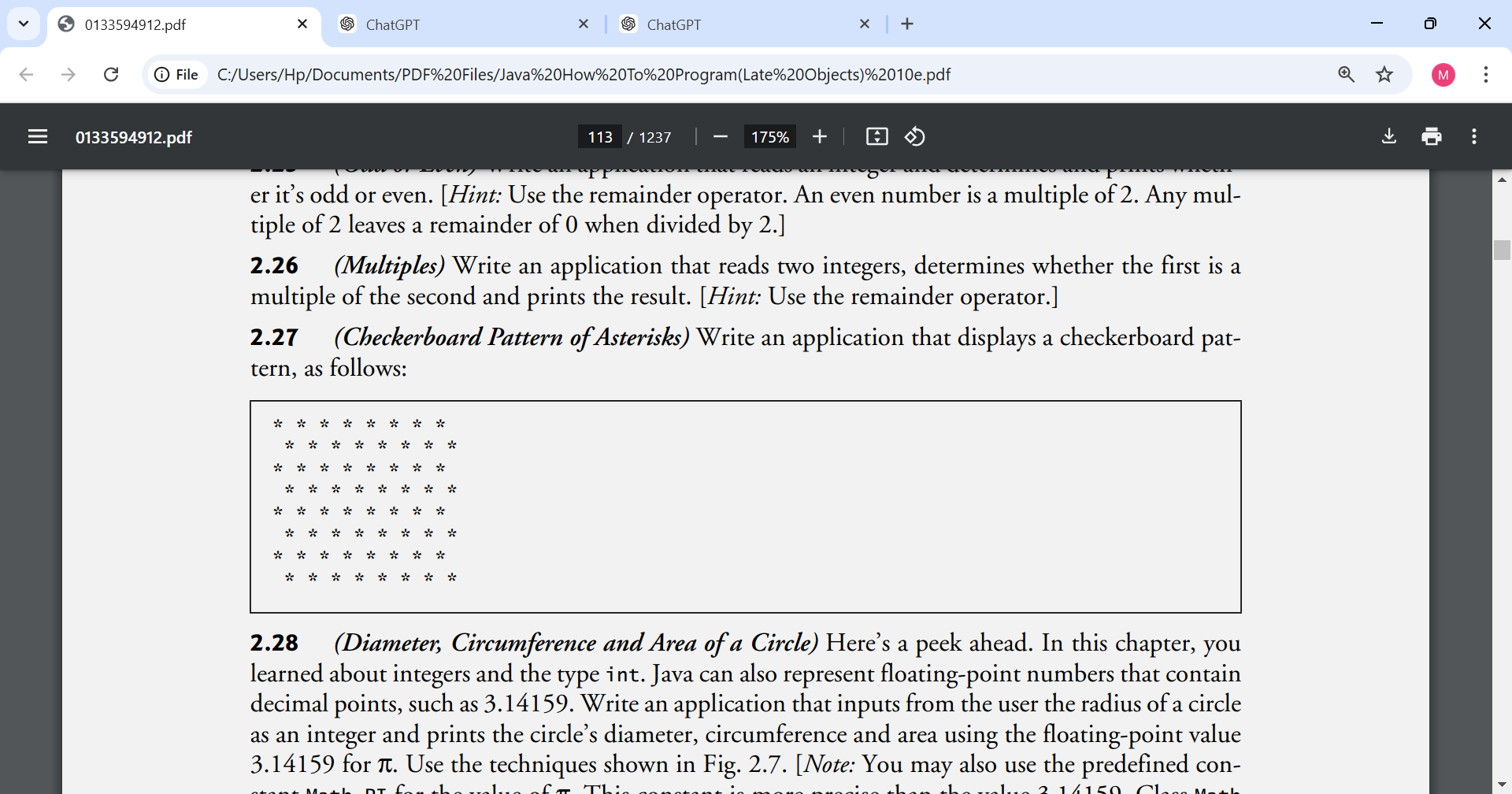
So it prints each string on its own line.

2.24 (Largest and Smallest Integers) Write an application that reads five integers and determines and prints the largest and smallest integers in the group. Use only the programming techniques you learned in this chapter.

2.25 (Odd or Even) Write an application that reads an integer and determines and prints whether it’s odd or even. [Hint: Use the remainder operator. An even number is a multiple of 2. Any multiple of 2 leaves a remainder of 0 when divided by 2.]

2.26 (Multiples) Write an application that reads two integers, determines whether the first is a multiple of the second and prints the result. [Hint: Use the remainder operator.]

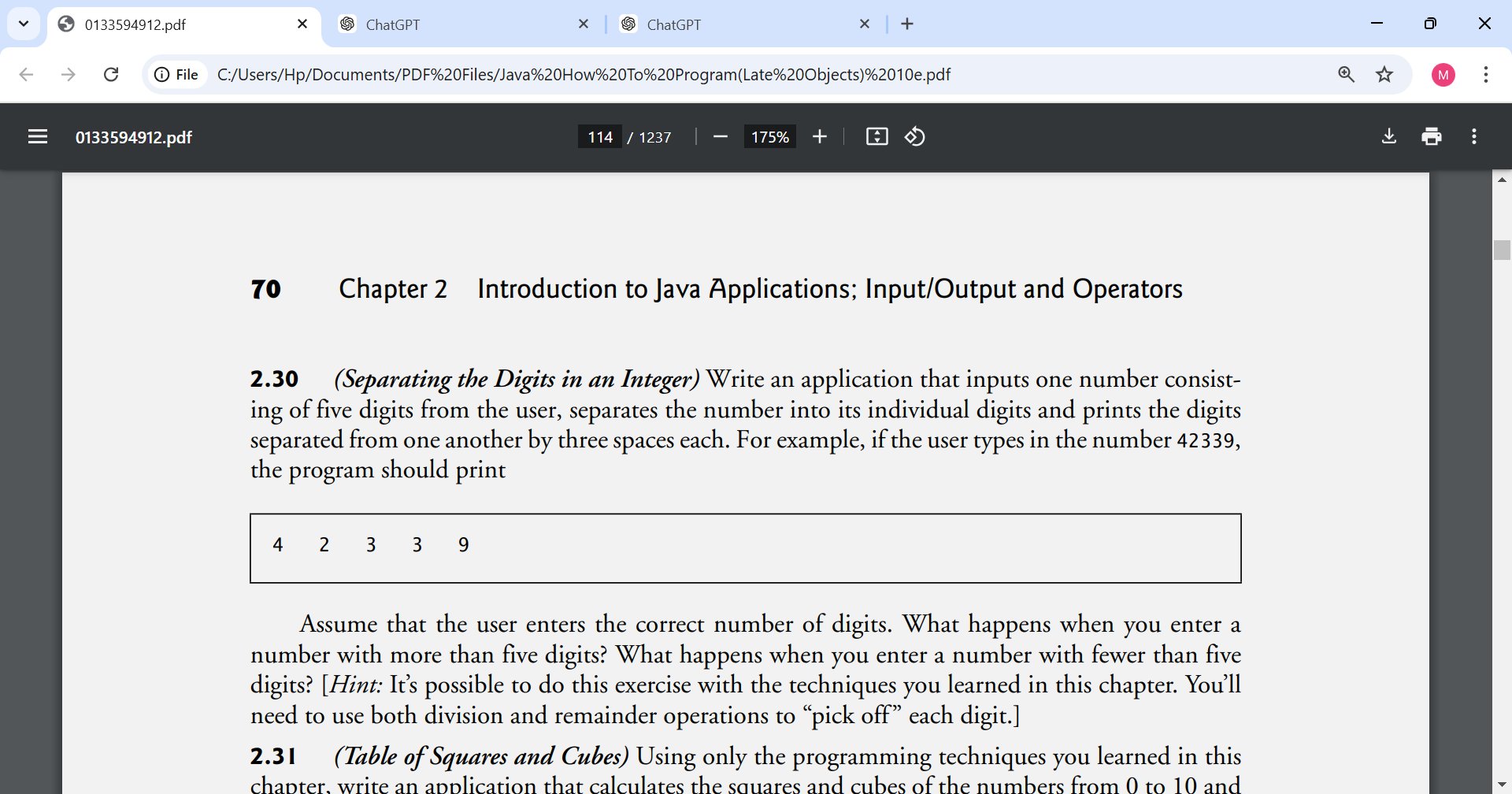
2.27 (Checkerboard Pattern of Asterisks) Write an application that displays a checkerboard pattern, as follows:



2.28 (Diameter, Circumference and Area of a Circle) Here’s a peek ahead. In this chapter, you learned about integers and the type int. Java can also represent floating-point numbers that contain decimal points, such as 3.14159. Write an application that inputs from the user the radius of a circle as an integer and prints the circle’s diameter, circumference and area using the floating-point value 3.14159 for π. Use the techniques shown in Fig. 2.7. [Note: You may also use the predefined constant Math.PI for the value of π. This constant is more precise than the value 3.14159. Class Math is defined in package java.lang. Classes in that package are imported automatically, so you do not need to import class Math to use it.] Use the following formulas (r is the radius): diameter = 2r circumference = 2πr area = πr2 Do not store the results of each calculation in a variable. Rather, specify each calculation as the value that will be output in a System.out.printf statement. The values produced by the circumference and area calculations are floating-point numbers. Such values can be output with the format specifier %f in a System.out.printf statement.

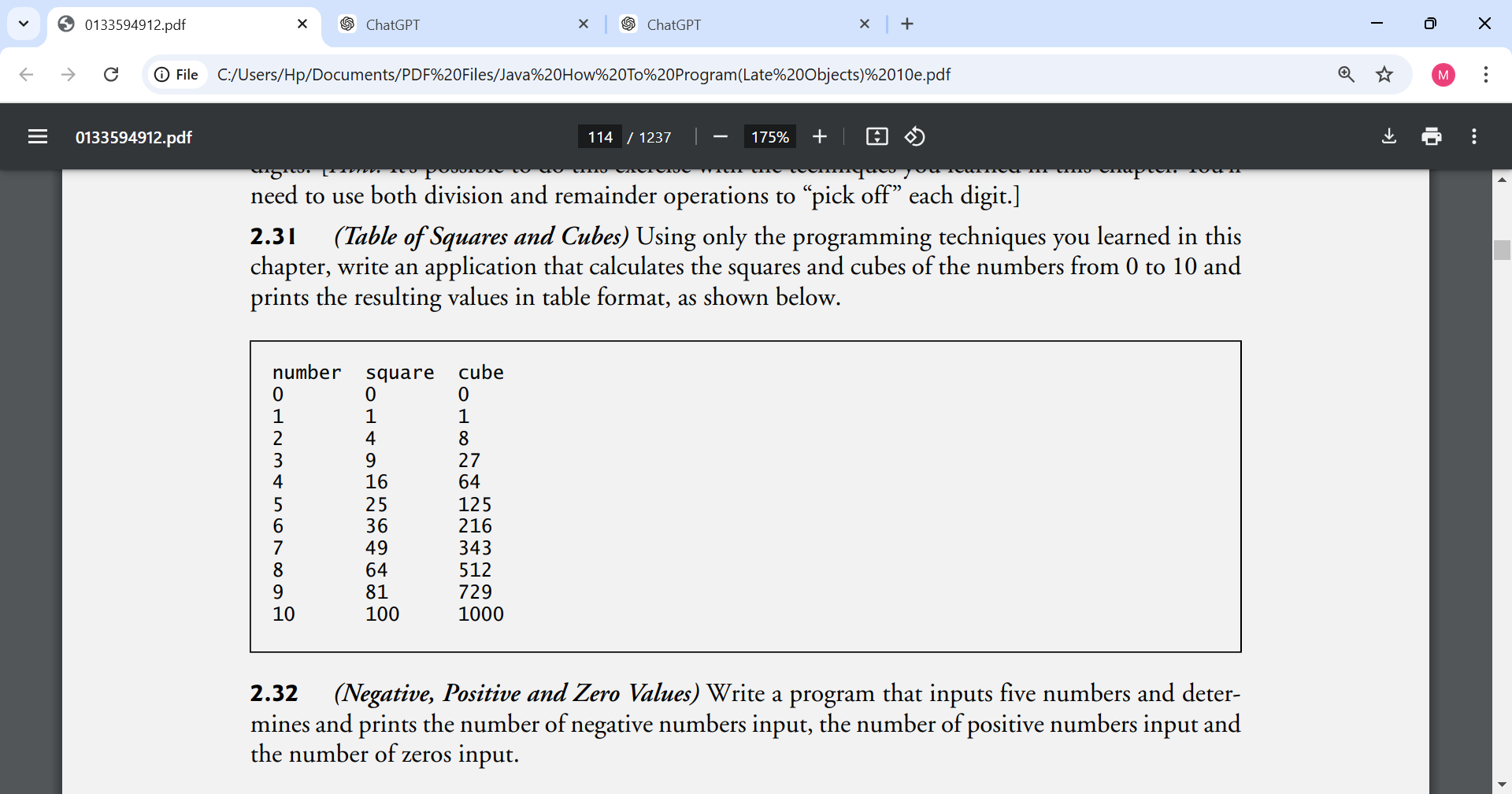
2.29 (Integer Value of a Character) Here’s another peek ahead. In this chapter, you learned about integers and the type int. Java can also represent uppercase letters, lowercase letters and a considerable variety of special symbols. Every character has a corresponding integer representation. The set of characters a computer uses together with the corresponding integer representations for those characters is called that computer’s character set. You can indicate a character value in a program simply by enclosing that character in single quotes, as in 'A'. You can determine a character’s integer equivalent by preceding that character with (int), as in (int) 'A' An operator of this form is called a cast operator. (You’ll learn about cast operators in Chapter 3.) The following statement outputs a character and its integer equivalent: System.out.printf("The character %c has the value %d%n", 'A', ((int) 'A')); When the preceding statement executes, it displays the character A and the value 65 (from the Unicode® character set) as part of the string. The format specifier %c is a placeholder for a character (in this case, the character 'A'). Using statements similar to the one shown earlier in this exercise, write an application that displays the integer equivalents of some uppercase letters, lowercase letters, digits and special symbols. Display the integer equivalents of the following: ABCabc012$\*+/ and the blank character.

2.30 (Separating the Digits in an Integer) Write an application that inputs one number consisting of five digits from the user, separates the number into its individual digits and prints the digits separated from one another by three spaces each. For example, if the user types in the number 42339, the program should print



Assume that the user enters the correct number of digits. What happens when you enter a number with more than five digits? What happens when you enter a number with fewer than five digits? [Hint: It’s possible to do this exercise with the techniques you learned in this chapter. You’ll need to use both division and remainder operations to “pick off ” each digit.]

2.31 (Table of Squares and Cubes) Using only the programming techniques you learned in this chapter, write an application that calculates the squares and cubes of the numbers from 0 to 10 and prints the resulting values in table format, as shown below.



2.32 (Negative, Positive and Zero Values) Write a program that inputs five numbers and determines and prints the number of negative numbers input, the number of positive numbers input and the number of zeros input.

Making a Difference

2.33 (Body Mass Index Calculator) We introduced the body mass index (BMI) calculator in Exercise 1.10. The formulas for calculating BMI are



Create a BMI calculator that reads the user’s weight in pounds and height in inches (or, if you prefer, the user’s weight in kilograms and height in meters), then calculates and displays the user’s body mass index. Also, display the following information from the Department of Health and Human Services/National Institutes of Health so the user can evaluate his/her BMI:



[Note: In this chapter, you learned to use the int type to represent whole numbers. The BMI calculations when done with int values will both produce whole-number results. In Chapter 3, you’ll learn to use the double type to represent numbers with decimal points. When the BMI calculations are performed with doubles, they’ll both produce numbers with decimal points—these are called “floating-point” numbers.]

2.34 (World Population Growth Calculator) Use the web to determine the current world population and the annual world population growth rate. Write an application that inputs these values, then displays the estimated world population after one, two, three, four and five years.

2.35 (Car-Pool Savings Calculator) Research several car-pooling websites. Create an application that calculates your daily driving cost, so that you can estimate how much money could be saved by carpooling, which also has other advantages such as reducing carbon emissions and reducing traffic congestion. The application should input the following information and display the user’s cost per day of driving to work:

a) Total miles driven per day.

b) Cost per gallon of gasoline.

c) Average miles per gallon.

d) Parking fees per day.

e) Tolls per day