**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** statement to make decisions.
* c) **//** begins an end-of-line comment.
* d) **Blank lines**, **spaces**, and **tabs** 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) False. Comments are ignored by the compiler and do not cause any text to be printed on the screen.
* b) True. All variables in Java must be declared with a specific data type.
* c) False. Java is case-sensitive, so number and NuMbEr are considered different variable names.
* d) True. The remainder operator (%) is used to get the remainder of a division, which requires integer operands.
* e) False. The arithmetic operators \*, /, and % have a 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. int c, thisIsAVariable, q76354, number;
* b) Prompt the user to enter an integer.

System.out.print("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. int value = input.nextInt();

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

System.out.println("This is a Java program");

* 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.

System.out.printf("This is a Java%nprogram%n");

* f) If the variable number is not equal to 7, display "The variable number is not equal to 7” if (number != 7) {

System.out.println("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"); **Correction:** The semicolon after the if condition is an error. It creates an empty statement, so the println statement will execute regardless of the condition.

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"); **Correction:** The operator >= should be used instead of =>. 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.

// Program to calculate the product of three integers

* b) Create a Scanner called input that reads values from the standard input.

Scanner input = new Scanner(System.in);

* c) Declare the variables x, y, z and result to be of type int. int x, y, z, result;

* d) Prompt the user to enter the first integer.

System.out.print("Enter the first integer: ");

* e) Read the first integer from the user and store it in the variable x. x = input.nextInt();

* f) Prompt the user to enter the second integer.

System.out.print("Enter the second integer: ");

* g) Read the second integer from the user and store it in the variable y. y = input.nextInt();

* h) Prompt the user to enter the third integer.

System.out.print("Enter the third integer: ");

* i) Read the third integer from the user and store it in the variable z. z = input.nextInt();

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

result = x \* y \* z;

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

System.out.printf("Product is %d%n", 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.** import java.util.Scanner;

public class ProductOfThreeIntegers {

public static void main(Stringargs) {

// Program to calculate the product of three integers

Scanner input = new Scanner(System.in); int x, y, z, result;

System.out.print("Enter the first integer: "); x = input.nextInt();

System.out.print("Enter the second integer: "); y = input.nextInt();

System.out.print("Enter the third integer: "); z = input.nextInt();

result = x \* y \* z;

System.out.printf("Product is %d%n", result);

}

}

**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 an **if statement**.
* c) Calculations are normally performed by **assignment** statements.
* d) The arithmetic operators with the same precedence as multiplication are **division (/)** and **remainder (%)**.
* 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.

System.out.print("Enter an integer: ");

* b) Assign the product of variables b and c to variable a. a = b \* c;

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

// This program performs a sample payroll calculation

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

* a) False. Java operators have precedence rules that determine the order of evaluation. ● b) True. These are all valid Java variable names.
* c) False. Operator precedence rules apply in Java.
* d) True. These are all invalid variable names because they start with a digit.

**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); **Output:** x = 2

* b) System.out.printf("Value of %d + %d is %d%n", x, x, (x + x)); **Output:** Value of 2 + 2 is 4
* c) System.out.printf("x ="); **Output:** x =
* d) System.out.printf("%d = %d%n", (x + y), (y + x)); **Output:** 5 = 5

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

* a) p = i + j + k + 7; (Modifies p)
* d) value = input.nextInt(); (Modifies value)

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

**Options with green are the correct java statements**

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

**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;
  + 1. 3 \* 6 (Result: 18)
    2. 18 / 2 (Result: 9)
    3. 7 + 9 (Result: 16)
    4. 16 - 1 (Result: 15) x = 15
* b) x = 2 % 2 + 2 \* 2 - 2 / 2;
  + 1. 2 % 2 (Result: 0)
    2. 2 \* 2 (Result: 4)
    3. 2 / 2 (Result: 1)
    4. 0 + 4 (Result: 4)
    5. 4 - 1 (Result: 3) x = 3
* c) x = (3 \* 9 \* (3 + (9 \* 3 / 3)));
  + 1. 9 \* 3 (Result: 27)
    2. 27 / 3 (Result: 9)
    3. 3 + 9 (Result: 12)
    4. 3 \* 9 (Result: 27)
    5. 27 \* 12 (Result: 324) 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.** ● a) Use one System.out.println statement.

* System.out.println("1 2 3 4");

* b) Use four System.out.print statements.

System.out.print("1 ");

System.out.print("2 ");

System.out.print("3 ");

System.out.print("4");

* c) Use one System.out.printf statement.

System.out.printf("%d %d %d %d%n", 1, 2, 3, 4);

**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).** import java.util.Scanner;

public class Arithmetic {

public static void main(Stringargs) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the first integer: "); int number1 = input.nextInt();

System.out.print("Enter the second integer: "); int number2 = input.nextInt();

int sum = number1 + number2; int product = number1 \* number2; int difference = number1 - number2;

int quotient = number1 / number2; // Integer division

System.out.printf("Sum: %d%n", sum);

System.out.printf("Product: %d%n", product);

System.out.printf("Difference: %d%n", difference);

System.out.printf("Quotient: %d%n", quotient);

}

}

**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".**

import java.util.Scanner;

public class ComparingIntegers {

public static void main(Stringargs) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the first integer: "); int number1 = input.nextInt();

System.out.print("Enter the second integer: "); int number2 = input.nextInt();

if (number1 > number2) {

System.out.printf("%d is larger%n", number1);

} else if (number2 > number1) {

System.out.printf("%d is larger%n", number2);

} else {

System.out.println("These numbers are equal");

}

}

}

**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.**

import java.util.Scanner;

public class SmallestLargest {

public static void main(Stringargs) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the first integer: "); int number1 = input.nextInt();

System.out.print("Enter the second integer: "); int number2 = input.nextInt();

System.out.print("Enter the third integer: "); int number3 = input.nextInt();

// Calculate sum, average, and product int sum = number1 + number2 + number3; int average = sum / 3;

int product = number1 \* number2 \* number3;

// Determine smallest and largest int smallest = number1; int largest = number1;

if (number2 < smallest) { smallest = number2;

} if (number3 < smallest) { smallest = number3;

}

if (number2 > largest) { largest = number2;

} if (number3 > largest) { largest = number3;

}

// Display results

System.out.printf("Sum: %d%n", sum);

System.out.printf("Average: %d%n", average);

System.out.printf("Product: %d%n", product);

System.out.printf("Smallest: %d%n", smallest);

System.out.printf("Largest: %d%n", largest);

}

}

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

(Note: It's difficult to create perfect shapes with asterisks in a text-based format, but here's an approximation)

public class DisplayShapes {

public static void main(String[] args) {

// Box

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

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

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

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

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

// Oval

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

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

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

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

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

// Arrow

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

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

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

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

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

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

// Diamond

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

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

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

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

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

}

}

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

The code System.out.printf("\*%n\*\*%n\*\*\*%n\*\*\*\*%n\*\*\*\*\*%n"); prints:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

2.20 What does the following code print?

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

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

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

It prints

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*

**2.21 What does the following code print?**

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

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

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

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

The code prints everything on the same line

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

**2.22 What does the following code print?**

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

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

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

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

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

It prints

\*\*\*

\*\*\*\*\*

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

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

It prints

\*

\*\*\*

\*\*\*\*\*

**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.

import java.util.Scanner;

public class LargestSmallest {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int largest = Integer.MIN\_VALUE;

int smallest = Integer.MAX\_VALUE;

// Read five integers

System.out.println("Enter five integers:");

for (int i = 0; i < 5; i++) {

int number = input.nextInt();

if (number > largest) largest = number;

if (number < smallest) smallest = number;

}

// Print results

System.out.printf("Largest: %d%n", largest);

System.out.printf("Smallest: %d%n", smallest);

input.close();

}

}

**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.]**

import java.util.Scanner;

public class OddOrEven {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Read an integer

System.out.print("Enter an integer: ");

int number = input.nextInt();

// Check if odd or even

if (number % 2 == 0) {

System.out.println("The number is even.");

} else {

System.out.println("The number is odd.");

}

input.close();

}

}

**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.]**

import java.util.Scanner;

public class Multiples {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Read two integers

System.out.print("Enter the first integer: ");

int num1 = input.nextInt();

System.out.print("Enter the second integer: ");

int num2 = input.nextInt();

// Check if the first is a multiple of the second

if (num1 % num2 == 0) {

System.out.printf("%d is a multiple of %d%n", num1, num2);

} else {

System.out.printf("%d is not a multiple of %d%n", num1, num2);

}

input.close();

}

}

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

public class CheckerboardPattern {

public static void main(String[] args) {

for (int i = 0; i < 8; i++) {

if (i % 2 == 0) {

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

} else {

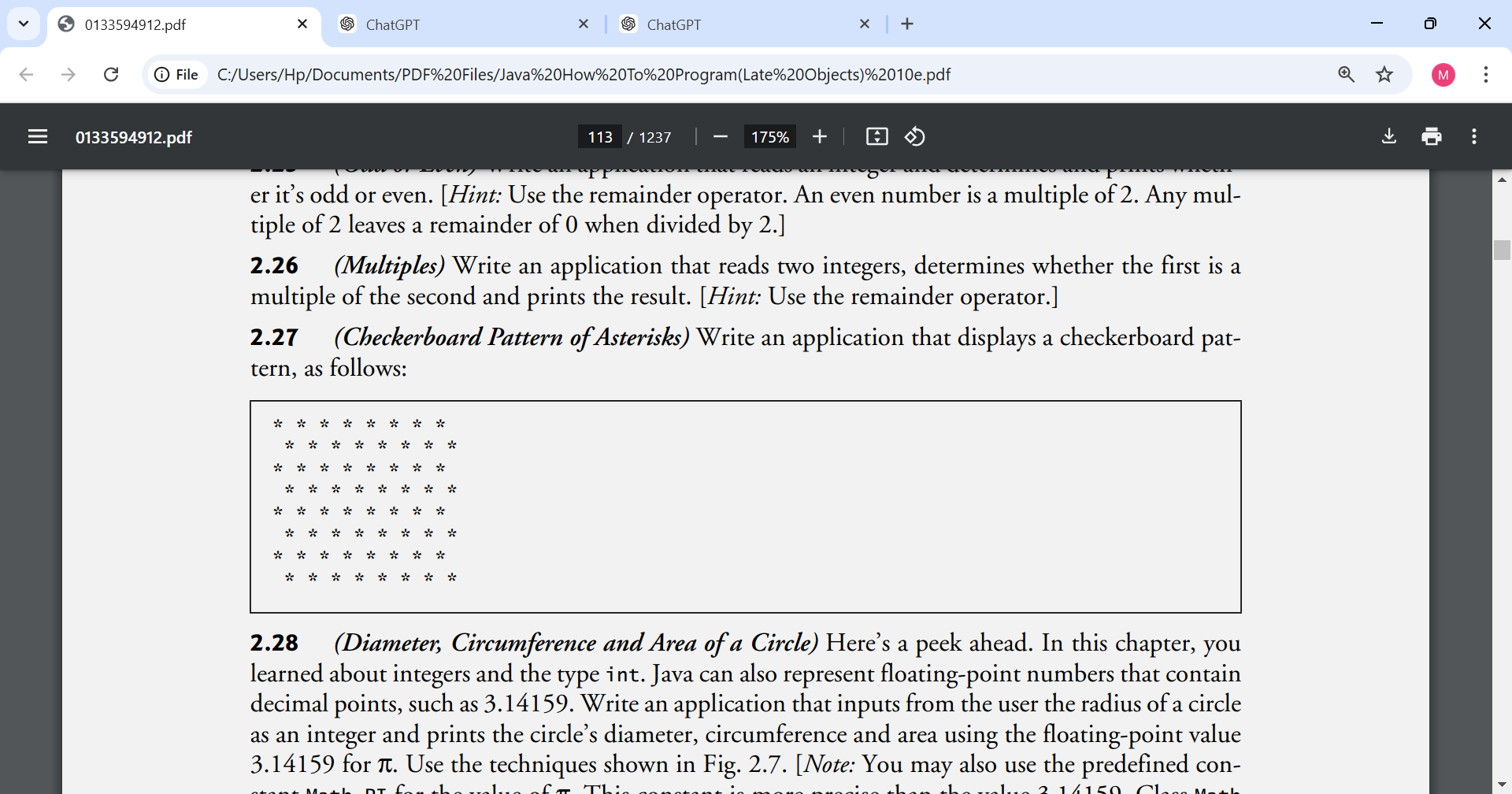
System.out.println(" \* \* \* \* \* \* \* \*");

}

}

}

}



**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.**

import java.util.Scanner;

public class CircleCalculations {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter the radius of the circle

System.out.print("Enter the radius of the circle as an integer: ");

int radius = input.nextInt();

// Print the diameter

System.out.printf("Diameter of the circle: %d%n", 2 \* radius);

// Print the circumference

System.out.printf("Circumference of the circle: %.2f%n", 2 \* Math.PI \* radius);

// Print the area

System.out.printf("Area of the circle: %.2f%n", Math.PI \* radius \* radius);

input.close();

}

}

**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.**

public class CharacterIntegerValues {

public static void main(String[] args) {

// Characters to display their integer equivalents

char[] characters = {'A', 'B', 'C', 'a', 'b', 'c', '0', '1', '2', '$', '\*', '+', '/', ' '};

System.out.println("Character\tInteger Equivalent");

// Loop through each character and display its integer value

for (char character : characters) {

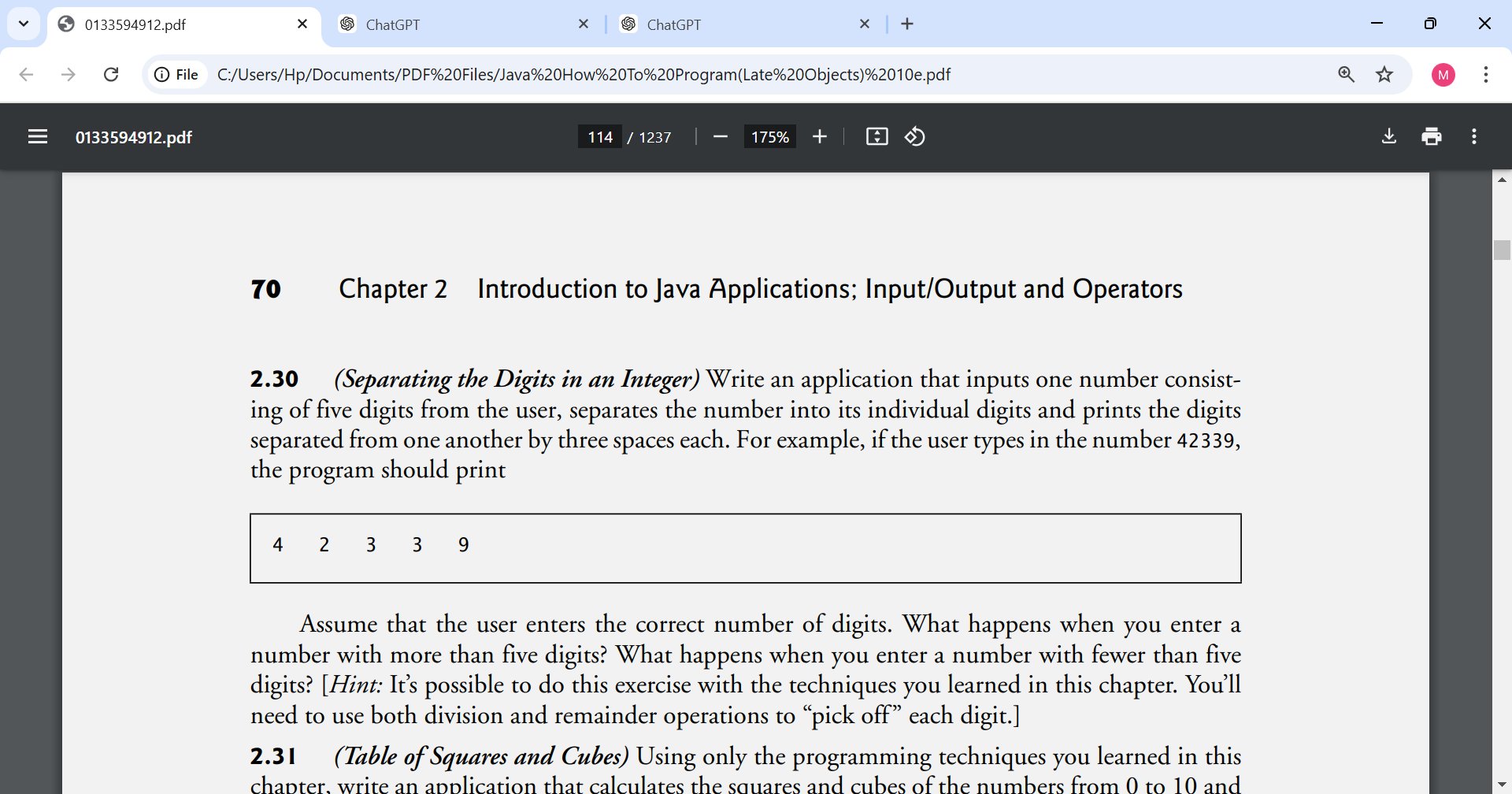
System.out.printf("%c\t\t%d%n", character, (int) character);

}

}

}

**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.]

import java.util.Scanner;

public class SeparateDigits {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter a five-digit number

System.out.print("Enter a five-digit number: ");

int number = input.nextInt();

// Extract and print each digit separated by three spaces

System.out.printf("%d %d %d %d %d%n",

(number / 10000), // First digit

(number / 1000) % 10, // Second digit

(number / 100) % 10, // Third digit

(number / 10) % 10, // Fourth digit

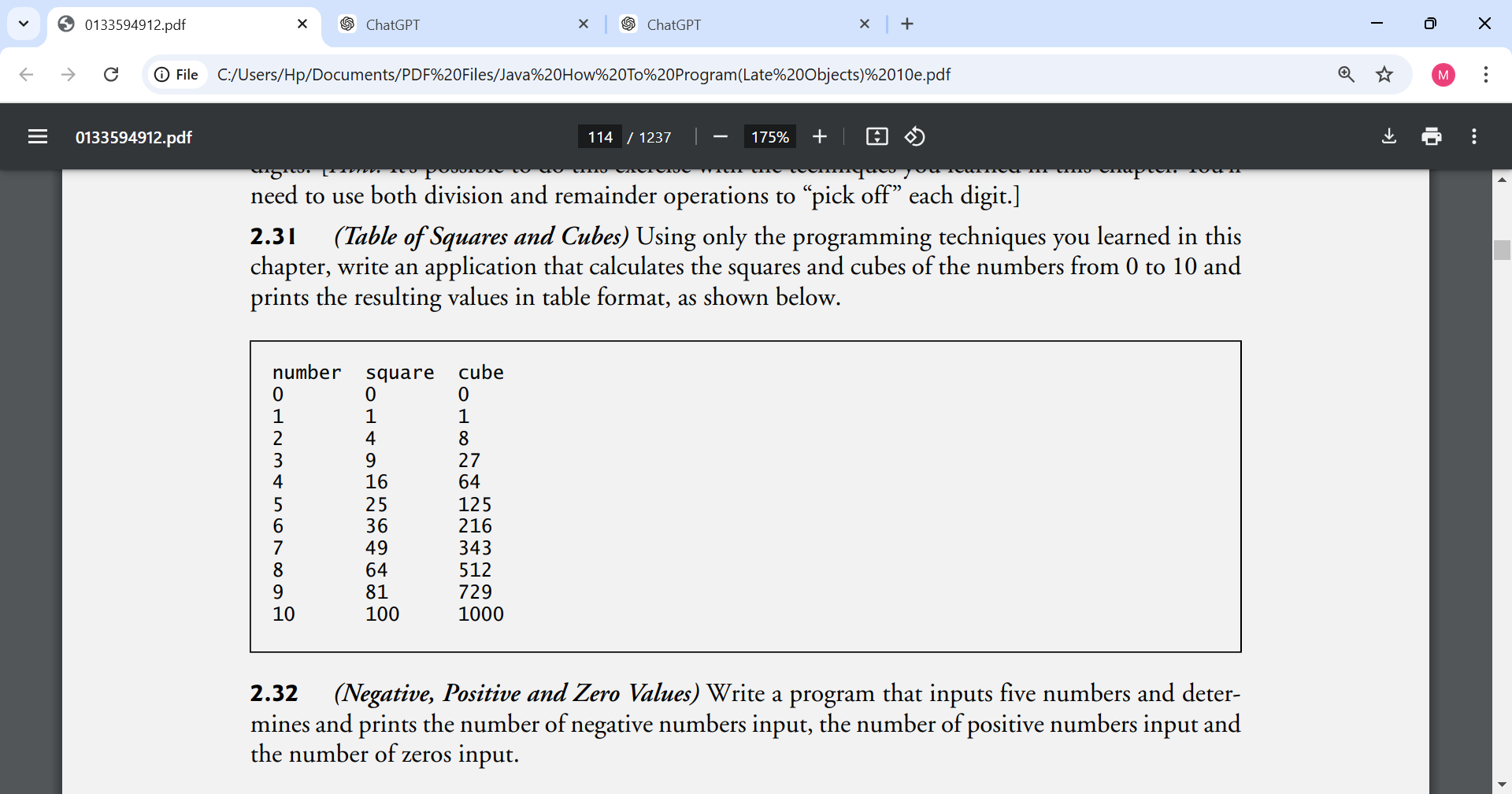
number % 10); // Fifth digit

input.close();

}

}

**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.**



public class SquaresAndCubes {

public static void main(String[] args) {

System.out.println("Number\tSquare\tCube");

// Loop through numbers from 0 to 10

for (int number = 0; number <= 10; number++) {

System.out.printf("%d\t%d\t%d%n",

number,

number \* number, // Square

number \* number \* number); // Cube

}

}

}

**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.**

import java.util.Scanner;

public class NumberCategorizer {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int positives = 0; // Counter for positive numbers

int negatives = 0; // Counter for negative numbers

int zeros = 0; // Counter for zero values

// Loop to input five numbers

System.out.println("Enter five numbers:");

for (int i = 0; i < 5; i++) {

int number = input.nextInt();

if (number > 0) {

positives++; // Increment positive counter

} else if (number < 0) {

negatives++; // Increment negative counter

} else {

zeros++; // Increment zero counter

}

}

// Display the results

System.out.printf("Number of positive numbers: %d%n", positives);

System.out.printf("Number of negative numbers: %d%n", negatives);

System.out.printf("Number of zeros: %d%n", zeros);

input.close();

}

}

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**



import java.util.Scanner;

public class BMICalculator {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user for weight and height

System.out.print("Enter your weight in kilograms: ");

double weight = input.nextDouble();

System.out.print("Enter your height in meters: ");

double height = input.nextDouble();

// Calculate BMI

double bmi = weight / (height \* height);

// Display the BMI result

System.out.printf("Your Body Mass Index (BMI) is: %.2f%n", bmi);

// Provide BMI categories

System.out.println("BMI Categories:");

System.out.println("Underweight: less than 18.5");

System.out.println("Normal weight: 18.5 - 24.9");

System.out.println("Overweight: 25 - 29.9");

System.out.println("Obesity: 30 or greater");

input.close();

}

}

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.**

**import java.util.Scanner;**

public class WorldPopulationGrowth {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user for the current population and growth rate

System.out.print("Enter the current world population (in billions): ");

double currentPopulation = input.nextDouble();

System.out.print("Enter the annual growth rate (as a percentage): ");

double growthRate = input.nextDouble();

// Display the estimated population for the next five years

System.out.println("Year\tEstimated Population (in billions)");

for (int year = 1; year <= 5; year++) {

currentPopulation += currentPopulation \* (growthRate / 100);

System.out.printf("%d\t%.3f%n", year, currentPopulation);

}

input.close();

}

}

**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

import java.util.Scanner;

public class CarPoolSavingsCalculator {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user for inputs

System.out.print("Enter total miles driven per day: ");

double milesDriven = input.nextDouble();

System.out.print("Enter cost per gallon of gasoline: ");

double costPerGallon = input.nextDouble();

System.out.print("Enter average miles per gallon: ");

double milesPerGallon = input.nextDouble();

System.out.print("Enter parking fees per day: ");

double parkingFees = input.nextDouble();

System.out.print("Enter tolls per day: ");

double tolls = input.nextDouble();

// Calculate daily driving cost

double gasolineCost = (milesDriven / milesPerGallon) \* costPerGallon;

double dailyDrivingCost = gasolineCost + parkingFees + tolls;

// Display the result

System.out.printf("Your daily driving cost is: $%.2f%n", dailyDrivingCost);

input.close();

}

}