* **Chapter Overview**

1. Computers execute sets of instructions called programs. Computers store information internally as sequences of 0s and 1s (binary numbers).
2. Programming and computer science deal with algorithms which are step-by-step descriptions for solving problems.
3. Java is a modern object-oriented programming language developed by Sun Microsystems, now owned by Oracle Corporation, that has a large set of libraries you can use to build complex programs.
4. A program is translated from text into computer instructions by another program called a compiler. Java’s compiler turns Java programs into a special format called Java bytecodes, which are executed using a special program called the Java Runtime Environment
5. Java Programmers typically complete their work using an editor called an Integrated Development Environment (IDE). The commands may vary from environment to environment, but the same three-step process is always involved:
   1. Type in a program as a Java class.
   2. Compile the program file.
   3. Run the compiled version of the program.
6. Java uses a command called system.out.println to display text on the console screen.
7. Written words in a program can take different meanings. Keywords are special reserved words that are part of the language. Identifiers are words defined by the programmer to name entities in the program. Words can also be put into strings, which are pieces of text that can be printed to the console.
8. Java programs that use proper spacing and layout are more readable to programmers. Readability is also improved by writing notes called comments inside the program.
9. That Java language has a syntax, or a legal set of commands that can be used. A Java program that does not follow the proper syntax will not compile. A program that does compile but that is written incorrectly may still contain errors called exceptions that occur when the program runs. A third kind of error is a logic or intent error. This kind of error occurs when the program runs but does not do what the programmer intended.
10. Commands in programs are called statements. A class can group statements into larger commands called static methods. Static methods help the programmer group code into reusable pieces. An important static method that must be part of every program is called main.
11. Iterative enhancement is the process of building a program piece by piece, testing the program at each step before advancing to the next.
12. Complex programming tasks should be broken down into the major tasks the computer must perform. This process is called procedural decomposition. Correct use of static methods aids procedural decomposition.

* **Section 1.1 Basic Computing Concepts**
* 1. Why do computers use binary numbers?
* To detect whether a circuit is powered on or off. This system is very Similar to Morse code as in when the circuit is powered on the computer interprets a 1, and vice versa when powered off the computer reads a 0. On / Off.
* 2. Convert each of the following decimal numbers into its equivalent binary number:
* a. 6
* b. 44
* c. 72
* d. 131

a. - 110

b. - 101100

c. - 1001000

d. – 10000011

* 3.
  1. 100
  2. 1011
  3. 101010
  4. 1001110
     1. 4
     2. 11
     3. 42
     4. 78
* 4. In your own words, describe an algorithm for baking cookies. Assume that you have a large number of hungry friends, so you’ll want to produce several batches of cookies.

Bake Cookies

* + - Collect Materials
      * Collect Money
        + Ask Friends
        + Go To Bank
      * Grocery Store
        + Find Cookies

Get Ingredients

Check Allergies

Spend Money

* + - Cook
      * Preheat Oven
        + Mix Ingredients
        + Wait 10 Mins

Start Baking

Let Cool

* + - Eat
      * Save
      * Stuff Face
* 5. What is the difference between the file MyProgram.java and the file MyProgram.class?
  + \*.java contains the source code written by the user, while \*.class contains java bytecodes and is generated during the compilation process.
* 6. Which of the following can be used in a Java program as identifiers?
  + Println
  + First-name
  + AnnualSalary
  + “hello”
  + ABC
  + 42isTheAnswer
  + for
  + sum\_of\_data
  + \_average
  + B4

**Chapter Summary**

1. Java groups data into types. There are two major categories of data types: primitive data and objects. Primitive types include int (integers), double (real numbers), char (individual text characters), and Boolean (logical values).
2. Values and computations are called expressions. The simplest expressions are individual values, also called literals. Some example literals are: 42, 3.14, ‘Q’, and false. Expressions may contain operators, as in (3 + 29) – 4 \* 5. The division operation is odd in that it’s split into quotient (/) and remainder (%) operations.
3. Rules of precedence determine the order in which multiple operators are evaluated in complex expressions. Multiplication and division are performed before addition and subtractions. Parentheses can be used to force a particular order of evaluation.
4. Data can be converted form one type to another by an operation called a cast.
5. Variables are memory locations in which values can be stored. A variable is declared with a name and a type. Any data value with a compatible type can be stored in the variable’s memory and used later in the program.
6. Primitive data can be printed on the console using the System.out.println method, just like text strings. A string can be connected to another value (concatenated) with the + operator to produce a larger string. This feature allows you to print complex expressions including numbers and text on the console.
7. A loop is used to execute a group of statements several times. The for loop is one kind of loop that can be used to apply the same statements over a range of numbers or to repeat statements a specified number of times. A loop can contain another loop, called a nested loop.
8. A variable exists from the line where it is declared to the right curly brace that encloses it. This range, also called the scope of the variable, constitutes the part of the program where the variable can legally be used. A variable declared inside a method or loop is called a local variable. A local variable can only be used inside its method or loop.
9. An algorithm can be easier to write if you first write an English description of it. Such a description is also called pseudocode.
10. Important constant values written into a program should be declared as class constants, both to explain their names and values and to make it easier to change their values later.

**Section 2.1 : Basic Data Concepts**

1. Which of the following are legal int literals?

22 1.5 -1 2.3 10.0 5. -6875309 ‘7’

* + 22, -1, -6875309, ‘7’
* 2. Trace the evaluation of the following expressions, and give their resulting values:
  1. 2 + 3 \* 4 – 6 = 8
  2. 14 / 7 \* 2 + 30 / 5 + 1 = 11
  3. (12 + 3) / 4 \* 2 = 6
  4. (238 % 10 + 3) % 7 = 4
  5. (18 – 7) \* (43 % 10) = 33
  6. 2 + 19 % 5 – (11 \* (5 / 2)) = -16
  7. 813 % 100 / 3 + 2.4 = 4.4\* (6.4)
  8. 26 % 10 % 4 \* 3 = 6
  9. 22 + 4 \* 2 = 30
  10. 23 % 8 % 3 = 1
  11. 12 – 2 – 3 = 7
  12. 6 / 2 + 7 / 3 = 5
  13. 6 \* 7 % 4 = 2
  14. 3 \* 4 + 2 \* 3 = 18
  15. 177 % 100 % 10 / 2 = 3
  16. 89 % (5 + 5) % 5 = 4
  17. 392 / 10 % 10 / 2 = 4
  18. 8 \* 2 – 7 / 4 = 15
  19. 37 % 20 % 3 \* 4 = 8
  20. 17 % 10 / 4 = 1
* 3. Trace the evaluation of the following expressions, and give their resulting values:
  1. 4.0 / 2 \* 9 / 2 = 9
  2. 2.5 \* 2 + 8 / 5.0 + 10 / 3 = 9.6
  3. 12 / 7 \* 4.4 \* 2 / 4 = 2.2
  4. 4 \* 3 / 8 + 2.5 \* 2 = 6
  5. (5 \* 7.0 / 2 – 2.5) / 5 \* 2 = 6
  6. 41 % 7 \* 3 / 5 + 5 / 2 \* 2.5 = 9.25\* (8.0)
  + 6 \* 3 / 5 + 5 / 2 \* 2.5
  + 18 / 5 + 5 / 2 \* 2.5
  + 3 + 5 / 2 \* 2.5
  + 3 + 2 \* 2.5
  + 3 + 5 = 8
  1. 10.0 / 2 / 4 = 1\* (1.25)

5 / 4 = 1.25

* 1. 8 / 5 + 13 / 2 / 3.0 = 3
  2. (2.5 + 3.5) / 2 = 3
  3. 9 / 4 \* 2.0 – 5 / 4 = 3
  4. 9 / 2.0 + 7 / 3 – 3.0 / 2 = 5
  5. 813 % 100 / 3 + 2.4 = 6.4
  6. 27 / 2 / 2.0 \* (4.3 + 1.7) – 8 / 3 2= 37
  7. 53 / 5 / (0.6 + 1.4) / 2 + 13 / 2 = 8.5
  8. 2.5 \* 2 + 8 / 5.0 + 10 / 3 = 9.6
* p. 2 \* 3 / 4 \* 2 / 4.0 + 4.5 – 1 = 4
* q. 89 % 10 / 4 \* 2.0 / 5 + (1.5 + 1.0 / 2) \* 2 = 4.8
* 4. Trace the evaluation of the following expressions, and give their resulting values:
* a. 2 + 2 + 3 + 4; = 11
* b. "2 + 2" + 3 + 4; = 2 + 234
* c. 2 + " 2 + 3 " + 4; = 2 2 + 3 4
* d. 3 + 4 + "2 + 2"; = 72 + 2
* e. "2 + 2 " + (3 + 4); = 2 + 2 7
* f. "(2 + 2) " + (3 + 4); = (2 + 2) 7
* g. "hello 34 " + 2 \* 4; = hello 34 8
* h. 2 + "(int) 2.0" + 2 \* 2 + 2; = 2(int) 2.042
* i. 4 + 1 + 9 + "." + (-3 + 10) + 11 / 3; = 14.73
* j. 8 + 6 \* -2 + 4 + "0" + (2 + 5); = 007
* k. 1 + 1 + "8 - 2" + (8 - 2) + 1 + 1; = 28 - 2611
* l. 5 + 2 + "(1 + 1)" + 4 + 2 \* 3; = 7(1 + 1)46
* m. "1" + 2 + 3 + "4" + 5 \* 6 + "7" + (8 + 9); = 123430717
* **Section 2.2: Variables**
* 5. What’s the correct syntax for declaring a real number variable named grade and initializing its value to 4.0?

1. int grade : 4.0;
2. grade = double 4.0;
3. double grade = 4.0;
4. grade = 4;
5. 4.0 = grade;

* 6. Imagine you are writing a personal fitness program that stores the user’s age, gender, height (in feet or meters), and weight (to the nearest pound or kilogram). Declare variables with the appropriate names and types to hold this information.

int age = 20;

boolean gender = male;

double height = 5.9;

int weight = 132;

* 7. Imagine you are writing a program that stores a student’s year (Freshman, Sophomore, Junior, or Senior), the number of courses the student is taking, and his or her GPA on a 4.0 scale. Declare variables with the appropriate names and types to hold this information.

String year = Freshman;

int courses = 1;

double gpa = 4.0;

* 8. Suppose you have an int variable called number. What Java expression produces the last digit of the number (the 1s place)?

1. number % 10

9. Fix the 9 mistakes.. yughh

public class Oops2

{

public static void main(String[] args)

{

int x;

System.out.println(“x is” x);

int x = 15.2; // set x to 15.2

System.out.println(“x is now + x”);

int y; // set y to 1 more than x

y = int x + 1;

System.out.println(“x and y are “ + x + and + y);

}

* }
* public class buggy
* {
* public static void main (String[] args)
* {
* double x = 0;
* System.out.println("x is " + x);
* // x gets 15.2
* x = 15.2;
* System.out.println("x is now " + x);
* // y gets 1 more than x
* double y = x + 1;
* System.out.println("x and y are " + x + " and " + y + " not to mention that x + y = " + (x + y));
* }
* }
* 10. Suppose you have an int variable called number. What Java expression produces the second-to-last digit of the number (the 10s place)? What expression produces the third-to-last digit of the number (the 100s place)?
  1. Number % 100 = 2nd to last digit (10s place)
  2. Number % 1000 = 3rd to last digit (100s place)
* 11. What are the values of a, b, and c after the following statements?
  + int a = 5;
  + int b = 10;
  + int c = b;
  + a = a + 1;
  + b = b – 1;
  + c = c + a

a = 6

b = 9

c = 16

* 12. What are the values of first and second at the end of the following code? How would you describe the net effect of the code statements in this exercise?

int first = 8; // first gets 8

int second = 19; // second gets 19

first = first + second; // first gets 27

second = first – second; // second gets 8

first = first – second; // first gets 19

* 13. Rewrite the code from the previous exercise to be shorter, by declaring the variables together and by using the special assignment operators (e.g., +-, -=, \*= and /=) as appropriate.

int first = 8, second = 19;

* first += second;
* second = first - second;;
* first -= second;
* 14. What are the values of i, j, and k after the following statements?
* int i = 2;
* int j = 3;
* int k = 4;
* int x = 9;

i = x - i - j;

j = x - j - k;

* k = x - i - k;
* j = 2
* k = 1
* 15. What is the output from the following code?
* int max;
* int min = 10;
* max = 17 – 4 / 10;
* max = max + 6;
* min = max – min;
* System.out.println(max \* 2); // 46
* system.out.println(max + min); // 36
* System.out.println(max); // 23
* System.out.println(min); // 13
* 16. Suppose you have a real number variable x. Write a Java expression that computes the following value y while using the \* operator only four times.
  + y = 12.3x^4 – 9.1 x^3 + 19.3x^2 + 4.6x + 34.2
* public class expression
* {
* // macro to give x a real number variable
* public static final double X = 0;
* public static void main(String[] args)
* {
* // breaks down function by operands
* double y0 = 12.3, y1 = 9.1, y2 = 19.3, y4 = 34.2;
* // sets a changable variable for x
* double x = X;
* for (double i = 0; i <= 4; i++)
* {
* x = Math.pow(X,i);
* if (i == 4)
* y0 \*= x;
* if (i == 3)
* y1 \*= x;
* if (i == 2)
* y2 \*= x;
* if (i == 1)
* y3 \*= x;
* }
* // final computation
* double y = y0 - y1 + y2 + y3 + y4;
* 17. The following program redundantly repeats the same expressions many times. Modify the program to remove all redundant expressions using variables of appropriate types.
* // calculate pay at work based on hours worked each day
* double hours = 4 + 5 + 8 + 4;
* System.out.println("\n\nMy total hours worked:");
* System.out.println(hours + "\n");
* // payrate
* double salary = 8.75;
* System.out.println("My hourly salary:");
* System.out.println(salary + "\n");
* // total income
* double pay = hours \* salary;
* System.out.println("My total pay:");
* System.out.println("$" + pay + "\n");
* // taxes
* System.out.println("My taxes owed:"); // 20% tax
* System.out.println(pay \* .20 + "\n");