Chapter 3 Introduction to Parameters and Objects

3.1 Parameters

1. A parameter is any of a set of characteristics that distinguish different members of a family of tasks.

2. A formal parameter is a variable that appears inside parentheses in the header of a method that is used to generalize the method’s behavior.

3. An actual parameter is a specific value or expression that appears inside parentheses in a method call.

4. Some people prefer to reserve the word “argument” for actual parameters and the word “parameter” for formal parameters.

5. Provide an example of a method call that uses a parameter. Add(3, 4)

6. Provide an example of a method header that uses a parameter. Public static void add(int I, int j)

7. What happens when Java executes a method call? The parameters are initialized.

8. Will the following statement compile and run? Why or why not?

testPhrase(String message);

No, because of the extra “String” keyword.

9. Will the following segment of code compile and run? Why or why not?

public static void testPhrase(message) {

System.out.println(message);

}

No, because there is a type required with the formal parameter

10. What is the output of the following code? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public class ParameterCode {

public static void main(String[] args) {

int a = 10;

int b = 5;

update(a, b);

System.out.println(a + “ “ + b);

}

public static void update(a, b) {

a--;

b++;

System.out.println(a + “ “ + b);

}

}

4 11

5 10

11. It is standard practice to insert a line break after an operator or parameter and indent the line that follows by twice the normal indentation width when a method header with parameters exceeds about 80-100 characters.

12. The basic rule is to use a class constant when you only want to change the value from execution to execution. If you want to use different values within a single execution, use a parameter.

13. Method Signature is the name of a method, along with its number and type of parameters.

14. Method Overloading is the ability to define two or more different methods with the same name but different method signatures.