# Unit 1 Discussion: ch1-4 Java Basics, ch6 User-Defined Methods

1. What did you find most confusing or difficult about what you read this week? You may also ask for help on a PA/CA or use of zyBook. (Note: be specific. For example, instead of “xx is hard”, identify the topic and illustrate with an example.)

2. Describe your current practice of documenting a program (such as commenting) and briefly explain how it has benefited or hindered your learning and/or project development. Description of real world practice is welcome.

3. Briefly explain how a Java source code file is processed before it can be executed? Include the terms compiler, [Java] bytecode, and [Java] virtual machine. If needed, do some research on your own.

4. (Multiple correct answers) Which of the following is a valid identifier? Be sure to explain why each item is/is not valid.

A. $343

B. class

C. 9X

D. current temp

E. radius

5. What is the ending value of z?

int x = 5;  
int y = 12;  
double z;  
z = (double)(y / x);

A. 0.0

B. 2.0

C. 2.4

D. 3.0

6. What is y's ending value?

int x;

int y = 0;

x = scnr.nextInt();

if (x = 20) {

y = 3;

}

A. Always 0 no matter what the input

B. Always 20 no matter what the input

C. 3 when the input is 20, else 0

D. Error: Compiler will not allow the expression x = 20

7. Which input value causes the loop body to execute a 2nd time, thus outputting "In loop" again?

String s = "Go";

while ((!s.equals("q")) && (!s.equals("Q"))) {

System.out.println("In loop");

s = scnr.next(); // input a string

}

A. "q" only

B. "Q" only

C. Either "q" or "Q"

D. "Quit"

8. What is the output?

int num = 10;

while (num <= 15) {

System.out.print(num + " ");

if (num == 12) {

break;

}

++num;

}

System.out.print("Done");

A. 10 Done

B. 10 11 Done

C. 10 11 12 Done

D. 10 11 12 13 14 15 Done

9. Which is the best stub for a method that calculates an item's tax?

A.

public static double ComputeTax(double itemPrice) {  
 double TAXRATE = 0.0675;  
 return itemPrice \* TAXRATE;  
}

B.

public static double ComputeTax(double itemPrice) {  
 double tax;  
 return tax;  
}

C.

public static double ComputeTax(double itemPrice) {  
 System.out.println("FIXME: Calculate tax");  
 return 0.0;  
}

D.

public static double ComputeTax(double itemPrice) {  
 double tax = 0.0;  
}

10. Which XXX tests the input value 4 for squareNum()?

public static int squareNum(int origNum) {

return origNum \* origNum;

}

public static void main(String args[]) {

System.out.println("Testing started");

XXX

System.out.println("Testing completed");

}

A. test(4);

B. squareNum(4);

C. assert(test(4)==16) : "4, expecting 16, got: " + test(4);

D. assert(squareNum(4)==16) : "4, expecting 16, got: " + squareNum(4);

11. What is the output?

public static void swapValues(int x, int y) {  
 int tmp = x;  
 x = y;  
 y = tmp;  
}   
  
public static void main(String args[]) {  
 int p = 4, q = 3;  
 swapValues(p, q);  
 System.out.print("p = " + p + ", q = " + q);  
}

A. p = 3, q = 3

B. p = 4, q = 3

C. p = 3, q = 4

D. Error: Argument names must match parameter names

12. The following program contains 7 mistakes. What are they? See how many you can identify.

public class Oops {

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;

y = int x + 1;

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

}

}

13. What will the following code print? Explain by tracing the execution.

for (int i = 1; i <= 6; i++) {

for (int j = 6; j >= 1; j--)

if (j <= i)

System.out.print(j + " ");

else

System.out.print(" " + " "); // 2 spaces

System.out.println();

}

14. Write a segment of code to calculate 0.1 + 0.2 + 0.3 + ... + 1.9 + 2.0. Declare necessary variables and use a loop.

15. Write a boolean method to determine if a given string contains a valid Missouri license number. Assume a valid license number is in the form of "LLDLDL" with L representing a letter and D representing a digit. Allow uppercase and lowercase letters. You may use regular expression related techniques, but only after you can solve the problem with the material we’ve covered.

Example valid license number strings: DJ3T3X, cc8D0P

Example invalid license number strings: D3J3TX (invalid character at specific index locations), PM5P (too short), BO1T4XJM (too long), U$8E4E (invalid character at specific index locations)

// Returns true if the string parameter contains a valid Missouri

// license number.

public static boolean isValid(String driverLicenseNum) {

/\* ADD your work here \*/

}

16. Primary U.S. interstate highways are numbered 1-99. Odd numbers (like the 5 or 95) go north/south, and evens (like the 10 or 90) go east/west. Auxiliary highways are numbered 100-999, and service the primary highway indicated by the rightmost two digits. Thus, I-405 services I-5, and I-290 services I-90.

Given a highway number, indicate whether it is a primary or auxiliary highway. If auxiliary, indicate what primary highway it serves. Also indicate if the (primary) highway runs north/south or east/west.

**Example Input | Example Output**

90 | I-90 is primary, going east/west.

290 | I-290 is auxiliary, serving I-90, going east/west.

0 (or >999) | 0 is not a valid interstate highway number (or the input n).

200 | 200 is not a valid interstate highway number. (It seems to be

a valid auxiliary highway num, but there is no 00 highway.)

/\*

\* TestHighwayNum.java

\* Decide whether an input number represents a valid primary or auxiliary highway number.

\*/

import java.util.Scanner;

public class TestHighwayNum {

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

int highwayNumber;

int primaryNumber;

System.out.print("Please enter a valid high way number (1~999): ");

highwayNumber = scnr.nextInt();

/\* ADD your code here. \*/

}

}

17. Write a method to compute and print Fibonacci numbers:

// prints out Fibonacci numbers F(0) ~ F(n) in the same line if n >= 0.

// otherwise print an error message

public static void printFib(int n) {

/\* ADD your work here \*/

}

The Fibonacci numbers are a sequence of integers in which the first two elements are 1, and each following element is the sum of the two preceding elements. The mathematical definition of each kth Fibonacci number is the following:

F(0) = 1

F(1) = 1

F(k) = F(k-1) + F(k-2), k >= 2

The first 12 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144

**Example run | Output**

printFib(-4)| Error: argument must be zero or positive

printFib(0) | 1

printFib(1) | 1 1

printFib(2) | 1 1 2

printFib(9) | 1 1 2 3 5 8 13 21 34 55

18. Write a program with total change amount in pennies as an integer input, and output the change using the fewest coins, one coin type per line. The coin types are Dollars, Quarters, Dimes, Nickels, and Pennies. Use singular and plural coin names as appropriate, like 1 Penny vs. 2 Pennies.

**Example Input | Output**

A negative int | Invalid input. Must be 0 or positive

0 | No change

45 | 1 Quarter

| 2 Dimes

/\*

\* This program prints proper changes for an input in pennies.

\*/

import java.util.Scanner;

public class CoinChange {

public static void main(String[] args) {

/\* ADD your work here. \*/

}

}