Final Practice Exam Solution

1) *(2 pts.)* Is Java an object-oriented, procedural, or functional programming language?

a) Procedural

b) Object-Oriented

c) Functional

2) (*2 pts.*) How many **distinct** values can be represented by a 7-bit value?

a) 64

b) 128

c) 256

d) 512

e) 1024

3) *(2 pts.)* Is it a compile-time or run-time error when a Java program throws an exception because it cannot open a file?

a) Run-time

b) Compile-time

4) *(2 pts.)* A _____ is a program that executes compiled Java code on a specific platform.

a) Java Virtual Machine

b) Java Compiler

c) Java Programming Manual

d) Eclipse Editor

e) None of the above

5) *(2 pts.)* Circle the letter of the Java statement that declares and allocates a 2-dimensional array of integers with **four rows** and **five columns**:

a) int array[4][5];

b) int array[5][4];

c) int array[ ] [ ] = new int [4][5];

d) int array[ ] [ ] = new int [5][4];

e) None of the above

6) *(2 pts.)* When calling a Java method, is the programmer required to explicitly provide the type information for each parameter?

a) Yes

b) No

7) *(2 pts.)* Which of the following can be returned from a Java method?

a) Any primitive type

b) Any class

c) 1-dimensional array

d) 2-dimensional array

e) All of the above

8) (*2 pts.)* Name 8 primitive data types and one Java class, identifying which is which.

Primitive types: char, byte, short, int, long, float, double, boolean

Classes: String, Scanner, PrintWriter, P5, R5, etc.

9) *(1 pt.)* Show the Java code to **declare** a variable of type String and **initialize** it to "Whatever".

String myString = "Whatever"; or
String myString = new String("Whatever");

10) *(1 pt.)* Show the Java code to **declare** and **allocate** an array with 497 elements of type double.

double dArray[] = new double[497];

11) *(2 pts.)* Show <u>one</u> line of Java code that **declares, allocates,** and **initializes** an array of type integer with exactly 4 elements whose values are 97, 33, 44, and 12, in that order.

int iArray[] = { 97, 33, 44, 12 };

12) *(1 pt. each)* Evaluate the following Java expressions, assuming **x** and **y** are both declared as integer variables and **x = 17** and **y = 6**:

a) x / y + 3 = **5**

b) x % y * 4 = **20**

c) x – y / 3 = **15**

d) x / 5 – y * 2 = **-9**

13) *(2 pts.)* Extend the code shown below to handle the **exception** that can occur when opening a file that does not exist. You do not have to add the import that is needed for the **exception**.

```
public void readFile(String filename) {

     try {

         File file = new File(filename);
         Scanner scan = new Scanner(file);

     } catch (IOException e) {
          System.out.println("Cannot open file: " + filename);
     }

}
```

14) *(2 pts.)* Show a line of code that shifts the binary literal 0b00000101 left by 2 bits into a variable named **byteValue** of type byte that has previously been declared.

byteValue = 0b00000101 << 2;

15) *(2 pts.)* What is the value of **byteValue** in the previous question after the shift operation has been performed? You can show the result as a binary literal or a decimal value.

0b00010100 or 20

16) *(4 pts.)* Write a Java **conditional** statement that implements the table below, where **grade** is an integer and **school** is a string. Both variables have been declared and **grade** has been initialized to a value greater than or equal to 1:

If **grade** is:	Set **school** to:
1, 2, 3, 4, 5	Elementary School
6, 7, 8	Junior High
9, 10, 11, 12	High School
> 12	College

```
if (grade <= 5)
     school = “Elementary School”;
else if (grade <= 8)
     school = “Junior High”;
else if (grade <= 12)
     school = “High School”;
else
     school = “College”;
```

17) *(4 pts.)* Write a Java **switch** statement for the table above that is identical to the statement in the previous problem.

```
if (grade <= 5)
     school = “Elementary School”;
else if (grade <= 8)
     school = “Junior High”;
else if (grade <= 12)
     school = “High School”;
else
     school = “College”;
```

18) *(2 pts.)* Using a Scanner object called **input** that has been instantiated and initialized, write code to read a string (without spaces), double, and integer from a file into variables you declare.

```
String s = input.next();
double d= input.nextDouble();
int i = input.nextInt();
```

19) *(2 pts.)* Convert the following into a **valid** Java expression: `15.2 ≤ limit ≤ 62.9:`

```
(15.2 <= limit && limit <= 62.9)
```

20) *(2 pts.)* Write a Java statement to assign the sixth character of the string variable **s** to a previously declared character variable named **c**.

```
c = s.charAt(5);
```

21) *(1 pt. each)* Write the output for the following Java code:

```
System.out.printf("%.4f%n", 45.13);          //  output:  45.1300
System.out.println("%.4f%n", 123.456789);    //  output:  123.4568
```

22) (*2 pts.)* Given a variable x declared as a double and initialized, show the code to cast and store the value of x into a variable y which is defined as a short. Truncation will occur, which is fine.

```
y = (short) x;
```

23) (*2 pts.*) Show the code to create an object named **myObject** by instantiating a class called **MyClass** and passing the constructor a String literal "hello" as the argument.

```
MyClass myObject = new MyClass("hello");
```

24)) *(5 pts.)* Write a **loop** that computes the average of an array of integers called **iArray** that has been previously declared and initialized. Store the result in a variable of type integer called **iAverage** that has been previously declared and initialized to zero.

```
for (int i = 0; i < iArray.length; i++) {
     iAverage += iArray[i];
}
iAverage /= iArray.length;
```

25) (*5 pts.)* Show the declaration for a method called **myMethod** that 1) is visible outside the class, 2) can only access class (static) variables, 3) returns an array of integers, and 4) accepts two parameters which are a String and double, in that order.

```
public static int[] myMethod(String param1, double param2);
```

26) *(2 pts. each)* Write out the output of the following code in the spaces provided:

```
public class FinalExam1 {

     public static void main(String[] args) {

          String S1 = "Java Programming";
          String S2 = " is taught";
          String S3 = " at Colorado State";

          int iSize = S3.length() + 3;
          System.out.println(iSize);

          char cChar = S1.charAt(10);
          System.out.println(cChar);

          int iIndex = S2.indexOf("p");
          System.out.println(iIndex);

          String sSubstr = S1.substring(1, 7);
          System.out.println(sSubstr);

          boolean bEquals = S2.equals(" is taught");
          System.out.println(bEquals);
     }
}
```

First line of output: **21**

Second line of output: **a**

Third line of output: **-1**

Fourth line of output: **ava Pr**

Fifth line of output: **true**

27) (2 pts. each) Write out the output of the following code in the spaces provided. **HINT**: Draw a diagram that shows the array contents.

```
public class FinalExam2

{

     public static void main(String[] args) {

          // Declare and allocate array
          double array[] = new double[5];

          // Initialize array

          for (int index = 0; index < array.length; index++)
              array[index] = (index * 3.0) + 0.5;

          // Print array information
          System.out.printf("%.2f%n", array.length - 2);
          System.out.printf("%.2f%n", array[2]);
          System.out.printf("%.2f%n",array[3]* 2.0);
          System.out.printf("%.2f%n", array[4]+0.45);
          System.out.printf("%.2f%n", array[5]));

     }

}
```

First line of output: **3**
Second line of output: **6.50**
Third line of output: **19.00**
Fourth line of output: **12.95**
Fifth line of output: **Exception (Out of bounds)**

28*(2 pts. each)* Show what is printed by the code shown below:

```
public class FinalExam3 {

     static double values[] = {0.0, 1.5, 3.0, 4.5, 6.0};
     static double variable = 5.0;

     public static void main(String[] args) {

          System.out.println(tripleNumber(10));
          System.out.println(tripleNumber(variable));
          System.out.println(tripleNumber(values[2]));
          System.out.println(values[2]);
          doubleArray (values);
          System.out.println(values[2]);

     }

     public static double tripleNumber(double param) {
          return param + param + param;
     }

     public static void doubleArray(double array[]) {
          for (int i = 0; i < array.length; i++)

               array[i] *= 2.0;
     }

}
```

First line of output:	**30.0**
Second line of output:	**15.0**
Third line of output:	**9.0**
Fourth line of output:	**3.0**
Fifth line of output:	**6.0**

29) *(2 pts. each)* Given the following code and a file named "data.txt" whose contents are listed on the next page, show what is printed. Assume that all required imports are defined above the code shown. If an exception occurs, write the word exception for that output line.

```
public class FinalExam4 {

    public static void main(String[] args) {

        int i0 = 2, i1 = 3;
        double d0 = 2.0, d1 = 3.0, d2 = 4.0;
        char c0 = '8', c1 = 'Z';
        String s0 = "", s1 = "";

        try {

            Scanner scan = new Scanner(new File("data.txt"));
            d0 = scan.nextDouble();
            i0 = scan.nextInt();
            c0 = scan.next().charAt(3);
            s0 = scan.next();
            System.out.println(d0 + ": " + i0);
            System.out.println(c0 + ": " + s0);

            if (scan.hasNextDouble())
                d1 = scan.nextDouble();
            if (scan.hasNextInt())
                i1 = scan.nextInt();

            c1 = scan.next().charAt(3);
            scan.nextLine(); // discard newline
            s1 = scan.nextLine();

            System.out.println(d1 + ", " + i1);
            System.out.println(c1 + ", " + s1);

            d2 = scan.nextDouble();
            System.out.println(d2);
            scan.close();

        } catch (Exception e) {
            System.out.println("Exception!");
        }
    }
}
```

Here are the contents of the file named "data.txt":

Line 0: **57.31 22**

Line 1: **Computer Science**

Line 2: **332211 1234.5**

Line 3: **Have a nice summer!**

Line 4: **double: 468.531**

First line of output: **57.31: 22**

Second line of output: **p: Science**

Third line of output: **332211.0, 3**

Fourth line of output: **4, Have a nice summer!**

Fifth line of output: **Exception**