**44-542 Object Oriented Programming Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exam 01 (100 points) KEY** *please print*

1. (5 pts) Suppose we have a **Dog** class with three private instance variables: **name** of type **String**, **breed** of type **String**, and **age** of type **int**. Complete the constructor below. The arguments should be assigned to the appropriate instance variables.

**public Dog(String name, String breed, int age)**

**{**

**this.name = name;**

**this.breed = breed;**

**this.age = age;**

**}**

1. (5 pts) Assume we have a **String** variable named **str1** that has been assigned a value. Write a single Java statement that prints **big** if **str1** has more than 8 characters and **small** otherwise. Use the conditional operator (**? :**) in your solution.

**System.out.println(str1.length() > 8 ? "big" : "small");**

1. (5 pts) Suppose we have a class **Student** with private instance variable **age**. Method **getAge** returns the age of the student:

**public int getAge()**

**{**

**return age;**

**}**

Write the Javadoc comments for this method.

**/\*\***

**\* Returns the age of the student**

**\* @return The age of the student**

**\*/**

1. (8 pts) Suppose we have the following declarations:

**int a = 5;**

**int b = 8;**

**int c = 11;**

Evaluate the following expressions:

* 1. **a + c / b % a + b - c - 2 \* a** : \_\_-7\_\_\_\_\_\_\_
  2. **a < b && !(b < c)**: \_\_false\_\_\_\_\_\_\_

1. (5 pts) Write a single Java statement to declare and initialize a Scanner object named **sc** to read from the file **myData.txt**. You may assume the file is in the root folder of the project.

**Scanner sc = new Scanner(new File("myData.txt"));**

1. (10 pts) Write the code segment to do the following: Create an ArrayList of integer values named **myNumbers**. Fill in the first 10 entries of this array list. For each entry, the value assigned should be five more than the index of the entry. So the entry at index 0 is 5, the entry at index 1 is 6, and so forth.

* Write only the code segment. Do not include any class or method headers.

**ArrayList<Integer> myNumbers = new ArrayList<Integer>();**

**for(int i = 0; i < 10; i++)**

**{**

**myNumbers.add(i + 5);**

**}**

1. (5 pts) Assume we have created an array of 10 int values named **myInts** and that we have filled the array. (Do NOT write the code to do this – we are assuming this has already been done.)

Write a *single* Java statement that prints the value stored at index 3 in **myInts**.

**System.out.println(myInts[3]);**

1. (4 pts) Find the output of the following code segment:

**OUTPUT**

**true**

**false**

**true**

**false**

**String str1 = "JavaIsFun";**

**String str2 = str1;**

**String str3 = new String ("JavaIsFun");**

**System.out.println(str2.equals(str3));**

**System.out.println(str2 == str3);**

**System.out.println(str1.equals(str2));**

**System.out.println(str1 != str2);**

1. (12 pts) Find the output of the following code segment:

**OUTPUT**

**outer = 5**

**inner = 5**

**num = 40**

**outer = 5**

**inner = 6**

**num = 44**

**outer = 5**

**inner = 7**

**num = 48**

**outer = 7**

**inner = 7**

**num = 56**

**outer = 7**

**inner = 8**

**num = 60**

**outer = 7**

**inner = 9**

**num = 64**

**for(int outer = 5; outer < 9; outer += 2)**

**{**

**for(int inner = outer; inner < outer + 3; inner++)**

**{**

**System.out.println("outer = " + outer);**

**System.out.println("inner = " + inner);**

**int num = inner + outer;**

**while(num < 35)**

**{**

**num \*= 2;**

**}**

**System.out.println("num = " + num);**

**}**

**}**

1. (15 pts) Trace the following code segment and find the output. Your trace must show a complete listing of every value assigned to x, y, and z.

**int x = 5;**

**OUTPUT**

**15 15 5**

**TRACE GOES HERE**

**x y z**

**5 20 0**

**7 19 -2**

**9 18 -4**

**11 17 -1**

**13 16 2**

**15 15 5**

**int y = 20;**

**int z = 0;**

**while(x < y)**

**{**

**x += 2;**

**y -= 1;**

**if(x + y > 27)**

**{**

**z += 3;**

**} else**

**{**

**z -= 2;**

**}**

**}**

**System.out.println(x + " " + y + " " + z);**

**Fill in the blanks (8 points – 4 points each).**

1. To copy any portion of one array into any segment of a second array, use the \_\_\_\_\_**arraycopy**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ method from the **System** class.
2. In the header for method **main**, the word \_\_\_\_\_**void**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_means the program does not return data to the JVM when it ends.

**Multiple choice (18 points – 2 points each).**  Write the letter corresponding to the BEST correct answer.

***Select only ONE answer for each question. If you select more than one answer, the entire question will be counted as wrong.***

1. Suppose we have written a Java program named **HelloWorld.java**. To compile in a command prompt window, use the command \_\_\_\_\_.
   1. **java HelloWorld.java**
   2. **javac HelloWorld.java**
   3. **java HelloWorld**
   4. **javac HelloWorld**
2. In a command prompt window, the java command does which of the following?
   1. translates the java bytecode into machine code and runs the program
   2. compiles the source code
   3. translates the Java bytecode into machine code
   4. compiles the source code, translates the java bytecode into machine code, and runs the program
3. Syntax errors are detected at \_\_\_\_\_.
   1. run time
   2. compile time
4. To store the state of an object, use \_\_\_\_\_.
   1. methods
   2. constructors
   3. comments
   4. attributes
5. Which of the following is true of constructors?
   1. a constructor is the same as a method – there is no difference between the two
   2. when you define a class, you must always supply at least one constructor
   3. a class may have several constructors
   4. all of the above are true
6. **System.out** is an object of the \_\_\_\_\_ class.
   1. **System**
   2. **PrintWriter**
   3. **Printer**
   4. **PrintStream**
7. If **age** is a private instance variable in class **Dog**, and method **getAge** returns the age of a **Dog** object, then **getAge** is a(n) \_\_\_\_\_ method.
   1. accessor
   2. mutator
8. Which of the following is true of primitive types?
   1. they are not objects
   2. they have no methods
   3. variables for primitive types store a reference
   4. all of the above are true
   5. only a) and b) are true
9. To direct output to a file, use the \_\_\_\_\_ class.
   1. **System**
   2. **PrintWriter**
   3. **Printer**
   4. **PrintStream**