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

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

* 1. (2 pts) Write a ***single*** Java statement that declares and initializes **myInts** to be an array of int values of size 100.

**int[] myInts = new int[100];**

* 1. (8 pts) Assume the array **myInts**, declared above, has been filled with values. Write the code segment necessary to print the values stored in **myInts**. 10 numbers should be printed on each line, with a single space separating pairs of numbers. Your code must work if we change the size of the array.

**for (int i = 0; i < myInts.length; i++)**

**{**

**System.out.print(myInts[i] + " ");**

**if ((i + 1) % 10 == 0)**

**{**

**System.out.println();**

**}**

**}**

* 1. (2 pts) Write a ***single*** Java statement that declares and initializes **myIntegers** to be an array list of Integer values.

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

* 1. (10 pts) Write a code segment to find and print the maximum value stored in the array list **myIntegers**.

**int max = myIntegers.get(0);**

**for(int myInteger : myIntegers)**

**{**

**if (myInteger > max)**

**{**

**max = myInteger;**

**}**

**}**

**System.out.println("Largest value: " + max);**

1. (2 pts) Write a ***single*** Java statement to declare a Scanner object named **in** to read from a file named **integers.txt**.

**Scanner in = new Scanner (new File("integers.txt"));**

1. (4 pts) Suppose we have a **Horse** class with private String variables and a constructor as shown here:

**private String regNum;**

**private String horseName;**

**public Horse(String regNum, String horseName)**

**{**

**this.regNum = regNum;**

**this.horseName = horseName;**

**}**

Write the code for a no-arg constructor that assigns **000** to **regNum** and **NoName** to **horseName**. The no-arg constructor must consist of the header line, plus a ***single*** statement in the body that invokes the two-arg constructor with the appropriate arguments.

**public Horse()**

**{**

**this("000", "NoName");**

**}**

1. Suppose we have an array list of integers named **myNumbers**. Assume at least 50 integers have been added to the array list. Write Java code to do each of the following. For parts a, b, and c, write a ***single*** Java statement to perform the indicated task.
   1. (2 pts) Print the value stored at index 37.

**System.out.println(myNumbers.get(37));**

* 1. (2 pts) Add 135 to the end of the array list.

**myNumbers.add(135);**

* 1. (2 pts) Replace the value at index 14 with the value 56.

**myNumbers.set(14,56);**

* 1. (4 pts) Write an enhanced for loop that will print each number in the array list. All numbers should be printed on the same line, with a single space separating pairs of numbers.

**for(int num : myNumbers)**

**{**

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

**}**

1. (2 pts) Write a ***single*** Java statement using the conditional operator (**? :**) that does the following. Test the value of an int variable **num** to see if it is less than 10. If it is, print the word **small**; otherwise, print the word **big**. Note: Assume the variable **num** has been declared and initialized. Do ***not*** write code to do this. Your code should be a single statement that tests the value of **num** and then prints the appropriate response.

**System.out.println(num < 10 ? "small" : "big");**

1. Suppose we have an array of 50 ints named **myNums**. Assume the array has been filled with values. Write Java code to do each of the following. For parts a and b, write a ***single*** Java statement to perform the indicated task.
   1. (2 pts) Print the value stored at index 37.

**System.out.println(myNums[37]);**

* 1. (2 pts) Replace the value at index 14 with the value 56.

**myNums[14] = 56;**

* 1. (4 pts) Write a traditional for loop (that is, do ***not*** use an enhanced for loop) that will print each number in the array. All numbers should be printed on the same line, with a single space separating pairs of numbers.

**for(int i = 0; i < myNums.length; i++)**

**{**

**System.out.print(myNums[i] + " ");**

**}**

1. (2 pts) Suppose we have executed the following statement in a Java program:

(statement 1) **String str = "34";**

We now try to assign **str** to an int variable using this statement.

(statement 2) **int myInt = str;**

This does not work and we get an “incompatible types” error.

Correct statement 2 above, by changing the right hand side of the expression. You cannot change statement 1, and you cannot change the left hand side of statement 2.

Hint: Casting will ***not*** work. You cannot cast a String variable to an int or Integer. They are inconvertible types.

**int myInt = Integer.parseInt(str);**

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

**String a = "Indiana Jones";**

**String b = "and the";**

**String c = "Temple of Doom";**

**System.out.println(c.substring(0, a.indexOf("a")));**

**System.out.println(**

**a.substring(0,5).concat(b.substring(b.indexOf(" ") + 1)));**

**OUTPUT**

**Temp**

**Indiathe**

1. (10 pts) Find the output of the following code segment.

**int x = 100;**

**OUTPUT**

**100 60**

**90 65**

**90 65**

**80 70**

**80 70**

**75 70**

**75 70**

**70 70**

**70 70**

**65 70**

**int y = 60;**

**while(x >= y)**

**{**

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

**if(x - y > 20)**

**{**

**x -= 10;**

**y += 5;**

**} else**

**{**

**x -= 5;**

**}**

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

**}**

1. (10 pts) Find the output of the following code segment:

**OUTPUT**

**35**

**50**

**10**

**16**

**65**

**95**

**for(int i = 0; i < 3; i++)**

**{**

**int num = (i + 5) \* 5;**

**if(num < 30)**

**{**

**num += 10;**

**} else if (num < 35)**

**{**

**num -= 20;**

**} else**

**{**

**num += 30;**

**}**

**System.out.println(num);**

**for(int j = 0; j < num; j += 10)**

**{**

**num += 3;**

**}**

**System.out.println(num);**

**}**

1. (6 pts) Find the output of the following code segment.

**OUTPUT**

**true**

**false**

**true**

**String str1 = "Hello";**

**String str2 = "Hello";**

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

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

**System.out.println(str1 == str3);**

**System.out.println(str1.equals(str3));**

**Multiple choice (20 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. The correct heading for method **main** is \_\_\_\_\_.
   1. **private void static main(String[] args)**
   2. **public void static main(String[] args)**
   3. **public static void main(String args)**
   4. **public static void main(String[] args)**
2. A Java application cannot accept arguments from the command line.
   1. true
   2. false
3. When creating Javadocs, if a method has multiple parameters, then multiple **@param** tags must be used.
   1. true
   2. false
4. Which of the following are true about array lists?
   1. an array list can grow as needed
   2. an array list can store primitive values
   3. in an array list, elements are stored at indexed positions.
   4. all of the above are true
   5. only a) and b) are true
   6. only a) and c) are true
5. The array list method that returns the current number of elements in the list is named
   1. **length**
   2. **size**
   3. **capacity**
   4. **current**
6. Suppose we have an array list of integers named **myValues**. Currently **myValues** has 10 numbers stored. Which of the following statements will throw an exception?
   1. myValues.add(10, 75);
   2. myValues.add(15, 33);
   3. myValues.remove(3);
   4. all of the above are illegal
   5. only a) and b) are illegal
7. An array is not an object.
   1. true
   2. false
8. Which of the following is a relational operator?
   1. %
   2. &&
   3. <=
   4. +
9. Which of the choices below gives the correct order of precedence, assuming the type of operator with the highest precedence is listed first?
   1. logical, relational, arithmetic
   2. arithmetic, relational, logical
   3. relational, logical, arithmetic
   4. arithmetic, relational, logical
10. The statement **Math.sqrt(2) \* Math.sqrt(2) == 2** returns \_\_\_\_\_.
    1. true
    2. false