

Your name: _____

You may not use any course materials in completing this test, only “the pages of your mind.”

The test consists of four sections.

There are 120 points possible, with 100 being the highest score.

Section 1 (30 points)

Fill in the blank (1 point per):

- _____ - What does CPU stand for?
- _____ - A constructor for a class is a type of what?
- _____ - How many constructors can a class have?
- _____ - What is the return type of `StringTokenizer.hasMoreTokens()`?
- _____ - What is the value of the expression `(4 < 3) || ('a' < 'b')`.
- _____ - Write the number 73 in binary.
- _____ - Write the number 1101100 in decimal.
- _____ - What type of method which calls itself.
- _____ - the number of swaps required for selection sort on a list of length **n**.
- _____ - What is the value of the expression `6 5 + 7 8 * +`
- _____ - What type of statements are `int x;` and `Rectangle r;`
- _____ - What type of statement is `x = 9;`
- _____ - In addition to the previous, what process occurs during the execution of the statement `r = new Rectangle();`

_____ - the return value of `str.indexOf("te")` given the previous statement
`String str = "Computer science is (kind of) fun";`

_____ - (2pt) What is the value of `n` after the execution of `n += n++`; as a function of the value `v` of the variable `n` before the statement execution.

_____ - (2pt) What is the value of `b` after the following code executes (assume `b` starts out greater than 0)? Write as a function of the previous value `v` of variable `b`.

```
int a;
for (a = b-1; a > 0; a --)
    b *= a;
```

Short answer:

(4pt each)

- What is a class? An object?
- What is an array in Java?
- What is overloading?

Section 2 (40 points)

(10 points)

What does the following code do, that is: what does this method return for a particular two arguments, `first <= last`? (Hint, try running on a couple of cases.)

```
public static int mystery(int first, int last)
{
    if (first > last)
        return 0;
    else if (first == last)
        return first;
    else
        return first + mystery(first + 1, last - 1);
}
```

(15 points) Consider the list {13, 9, 15, 2}. Show the list after each iteration of the outer loop for bubble, selection, and insertion sorting algorithms.

Bubble
13 9 15 2

Insertion
13 9 15 2

Selection
13 9 15 2

(15 points)

part a. 10pts. Consider the following list

3 14 19 35 43 55 63 89 91 95 110

How many comparisons are required to determine whether the following items are in the list or not (both linear and binary search)? Assume that you do not know that the list is ordered for the linear search.

- | | | |
|-------|-----------------|-----------------|
| a. 19 | Linear search : | Binary search : |
| b. 45 | Linear search : | Binary search : |
| c. 63 | Linear search : | Binary search : |
| d. 3 | Linear search : | Binary search : |
| e. 95 | Linear search : | Binary search : |

part b. 5 points. Assume that inserting a new element in this list is SEP (somebody else's problem). Describe an efficient scheme for inserting a new element in the list in the appropriate place.

Hint: you can use a method like insert(n, i) which inserts the new n at index i.

(You're more than half way there.)

Section 3 (40 points)

(20 points)

Provide list after the execution of the following method of LinkedList given the call "list.whatDoIDo(30);", if list.head originally references the list

4 -> 21 -> 22 -> 42 -> 25 -> 8 -> 34 -> 10 -> 29 -> 50 -> null

Alternatively, show the list at every check of the while loop condition.

```
public void whatDoIDo(int min)
{
    Node prev, cur;

    Node nnode = new Node(-10);
    nnode.next = head;

    prev = nnode;
    cur = head;

    while (cur != null)
    {
        if (cur.getVal() < min)
        {
            if (cur == head)
            {
                prev = head;
                head = head.next;
                cur = head;
            }
            else
            {
                prev.next = cur.next;
                cur = cur.next;
            }
        }
        else
        {
            prev = prev.next;
            cur = cur.next;
        }
    }
}
```

QUESTION: (5 of the 20 points) What does this method do?

(20 points)

a.(5pts) Provide the header for a method that takes an array of ints and returns the percentage of elements that are perfect squares. A perfect square is an integer number whose square root is also an integer number. (You have to give this method a name.)

b.(5pts) How would you check to see whether an element of the array (say, at index *i*) is indeed a perfect square? (Use the method `double Math.sqrt(double x)` to compute a square root.)

c.(10pts) Assume that *b* is solved with the boolean `CONDITION`. Write the body of this method.

Section 4 (10 points)

1.a. (5pts) Declare and instantiate an array of length 100 of integers and initialize the elements with their respective indices into the array.

1.b. (5pts) Compute the sum of the elements of the array and store this in a variable called `sum`. Or alternatively, tell me what the sum of the elements of the array is.

Thank you all.