

University of Illinois at Urbana-Champaign
 Department of Computer Science
Second Examination
 CS 125 Introduction to Computer Science
 90 minutes permitted

First name: _____ Last name: _____	AYA Tues 09:00am-10:50 AYB Tues 11:00am-12:50 AYC Tues 01:00pm-02:50 AYD Tues 03:00pm-04:50 AYE Tues 05:00pm-06:50 AYF Tues 07:00pm-08:50 AYG Wed 09:00am-10:50 AYH Wed 11:00am-12:50 AYK Wed 01:00pm-02:50 AYI Wed 03:00pm-04:50 AYJ Wed 05:00pm-06:50
NetID: _ _ _ _ _ @ illinois.edu <i>(please write legibly)</i>	
Discussion Section: AY_____ (or day+time) (We will return your exam manuscript to you in this section)	

- This is a closed book and closed notes exam. No electronic aids are allowed. You must turn this exam booklet in before leaving.
- You should have 9 sheets total including one scratch sheet The last sheet is scratch paper; you may detach it while taking the exam, but must turn it in with the exam when you leave.
- Carefully read the specification given for each class. You may not write, or assume the existence of, any unspecified additional methods or constructors.
- The proctors will not answer technical questions, only clarify English words for non-native speakers. If you believe a question contains an error or is ambiguous, write down your assumptions and answer accordingly.
- Unless we say otherwise in the specific problem, you can assume all values entered by the user will be acceptable input for that program.
- For full marks correct syntax is required: Ensure all statements include a semicolon and the correct use of upper/lower case, single quotes and double quotes. However a syntactically correct but flawed program will earn a low score i.e. semantics and correctness is more important than minor typos and syntax errors.

Problem	Points	Score	Grader
1	15		
2	16		
3	20		
4	18		
5	18		
6	13		
Total	100		

1. Short Answers – 15 points (3 points each)

(a) What is printed by the last line of the following Java code? Assume that the constructor initializes the private instance variables and that no compile or runtime errors occur. You do not need to write an explanation.

```
Student s1, s2, s3;
s1 = new Student ("A", 5,4);
s2 = new Student ("B", 5,4);
s3 = new Student ("C", 2,3);
s1 = s2;
s2 = s3;
s3 = s1;
System.out.print(s3 == s2);
System.out.print(s1 == s2);
```

Write exactly the program output here and nothing else: _____

(b) Briefly explain why no Tweet objects are created in the following code.

```
Tweet[] t = new Tweet[10];
TextIO.putln(t == null);
```

Write your explanation here: _____

(c) Is Java's String method *indexOf* an instance method or class method?

Write "instance" or "class" here: _____

(d) Your friend incorrectly claims Point objects are immutable (see class definition below). Write the line numbers of all of the lines inside the Point class definition that cause Pt objects to be mutable.

```
public class Point {
2  public int x,y;
3  public String asLabel() {return "("+x+", "+y+"): "+count;}
4  public int getX() {return x;}
5  public int getY() {return y;}
6  public void increment() { x++; y--; }
7  public boolean isValid() {return y>0; }
8  public void setValues(int[] arr) {x=arr[0];y=arr[1];}
9  public Point(int xx, int yy) { x=xx; y=yy; count++;}
10 public Point(Point src) { x=src.x; y=src.y; count++;}
11 public static int count=0;
}
```

Mutability due to (Line number(s)) : _____

(e) Using the Point class above, what will the following program print when executed? Also determine the new output if the 'static' modifier is deleted from 'int count' (line 11) and added to 'int x,y' (line 2)

<pre>Point a = new Point(10,20); Point b = new Point(30,40); b.increment(); System.out.println(a.asLabel()); System.out.println(b.asLabel());</pre>	Original output	Output after lines 2 & 11 changed

2. My Sketch items. – 16 points

You're working on a interactive sketch program for a mobile phone. To create a sketch, you record a sequence of touch points made by the user. The user may revisit the same position on the screen so points do not have unique (x,y) positions. Complete the `SketchList` class. Use the (unmodified) `Point` class as shown in Q1.

```
public class SketchList {
    private Point[] list = new Point[0];

    public int getSize()                {return list.length; }

    public Point getPoint(int index)    {return list[index]; }

    public void add(Point p) {
        Point[] tmp = new Point [list.length + 1];
        for(int i =0;i<list.length;i++)
            tmp [i] = list[i];
        tmp [tmp.length -1] = p;
        this.list = tmp;
    }
    /* Returns a new SketchList object with points from this original list at the requested (x,y) position. Hint: Use
    Point's getX,getY methods and the above add method to construct the result. */
    public SketchList find(int x, int y) {

    }

    /* Returns the first (zeroth index) item in the list and also removes it from the list i.e. the list size is reduced by
    one. If the list is already empty this method returns null and does not modify the list. */
    public Point removeFirst() {

    }

    /* Returns true iff any points in this list are at position (0,0), false otherwise. Use Sketchlist's find and getSize
    methods, specified above, to implement this method with minimal code (one or two lines). Assume the find
    method works as specified above. */
    public boolean anyPointsAtOrigin() {

    }

} // end class
```

3. I took CS125 and all I got was this startup company and nerdy friends – 20 Points

You are working on a photo manipulation program for a camera phone. Your startup partner has started the following code that processes a source image (*src*) and returns a new output image of a different size. Images are represented as a 2D array. The first array index represents the column, the second the row. Thus *array[x][y]* represents the red-green-blue pixel color at (x,y) encoded as an integer (similar to Photoscoop).



(a) Write code below so that `extract` returns an sub-image, represented as a 2D array. Return the pixels in the rectangle that starts at (*x1,y1*) and is of size (*w* by *h*) pixels.

- If the *src* parameter is null return a 2D integer array of size 1x1.
- Pixels in the output image that do not map to a valid source pixel should be grey (rgb value 0x808080).
- You may assume *x1,y1,h,w* are non-negative (i.e. ≥ 0).

```
public static int[][] extract(int[][] src, int x1,int y1,int w, int h) {
```

```
}
```

The expression, "`(rgb >> 8) & 0xff`" extracts the green color component value as an integer (0..255) by shifting right 8 bits then masking out the result to just the lowest 8 bits. For parts (b) and (c) you will use similar expressions to extract red and blue components as integers (0..255). The blue component (0..255) is packed in the lowest 8 bits of the pixel (bits 0 to 7), the green component in bits 8 to 15 and the red component is packed in bits 16 to 23.

(b) Complete the description of the following image transformation. Circle the correct responses.

```
if((src[x][y] & 0xff)>200) output[x][y]=0xff0000; else output[x][y]=0x00ff00;
```

"For each pixel, when the { Red / Green / Blue / Grey / White } component is greater than 200,

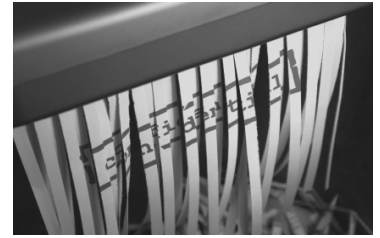
the output is { Red / Green / Blue / White / Black } otherwise it is { Red / Green / Blue / White }"

(c) Write an expression that returns true if the blue component is greater than the red component for 'rgb' – the color encoded as an integer of a single pixel.

```
public static boolean moreBlueThanRed(int rgb) {
    return
}
```

4. White-collar crime: Testing the de-shredder – 18 Points

At the FBI we're configuring a new piece of software that claims it can recreate a document after it's been shredded into pieces. To evaluate its effectiveness we will create test pictures by scrambling the order of the artificially generated vertical strips ('shreds'). I've already implemented a 'Strip' class. My *strip* class represents the image of a single vertical paper shred. It implements a copy constructor and a *getId()* method that returns the unique integer for each strip:



```
public class Strip { // Represents a single vertical strip/paper shred
    private int id;
    public Strip (Strip src) { this.id = src.id; /* Initialization of other variables not shown */}
    public Strip (int id) { this.id = id;}
    public int getId() { return this.id;}
    ... other instance variables and methods not shown
}
```

Complete the class *Document* according to the following specification. Do not create any other additional constructors, instance or class methods or instance or class variables – only create the constructors, methods and instance variables specified below.

1. A private array of type *Strip* named '*pieces*'.
2. A public constructor that takes a reference to an array of *Strip* objects and initializes *pieces* to refer to it. (Do not create a new array object, do not create any new Strips).
3. A public copy constructor that takes a reference to a *Document* object and uses *Strip*'s copy constructor to implement a *deep* copy.
4. A public method named *shuffle* that takes an integer 'N' and does not return anything. Perform N pairwise swaps to randomly shuffle the pieces to a new random ordering. A pairwise swap, simply exchanges two entries in the array, for example {6,7,8,9}->{6,9,8,7}. Use *Math.random()*, which returns a random double value between 0 and almost 1, to randomly select which array entries to swap.
5. A public method named '*score*' that takes no parameters and returns an integer. Use *Strip*'s *getId()* method to return the number of strip pieces that have an id value *one more than the previous strip's* id value. For example, this method will return 3 for the id sequence {1,2,3,6,4,5}. Assume all array references are valid i.e. non-null pointers to *Strip* objects.

```
public class Document {
```

5. My Dreshredding Simple Genetic Algorithm is too good for CSI – 18 Points

The evidence is shredded into 25 vertical strips. Create the following class methods in `Deshredder.java`. Do not create any other methods or instance variables or class variables. You may call other specified methods in this class, the `Strip` and `Document` classes (see Q4), and you may create temporary variables.

1. Create a public class method named `'create'` that takes no parameters and returns a new `Document`:
 - a. First create an array of 25 `Strip` objects for use by the new `Document`. Create each shred with a unique identifier (0... 24), then initialize a new document with this array.
 - b. Before returning the document, use the `Document`'s `shuffle` method to ensure all of the document's strips are in a random sequence.
2. Create a public class method named `'mutate'` that takes an array of `Document` objects (as specified in Q4) and two integers `'i'` and `'j'`. Do not return anything. Assume `'i'` and `'j'` refer to valid non-null array entries.
 - a. Replace the j^{th} entry with a new, *deep* copy of the i^{th} document.
 - b. Also use call `shuffle` (see Q4) on this new j^{th} entry to perform one random swap.
3. Create a public class method called `'main'` that takes a string array and does not return anything.
 - a. Use the `create` method above to create 10000 Documents and store them in an array.
 - b. Print out the `array index` of a document with the highest score of all documents. You only need to print one index, even if multiple documents have the same highest score.

```
public class Dreshredder {
```

```
} // end of class
```

6. Concepts and code snippets – 13 Points (Last Question)

(a) Write one line after the following code: Use variable 'ss' to cause a *NullPointerException* to be thrown.

```
String[] ss = new String[10];
```

(b) Identify the following use of three data structures a,b and c, as examples of a stack, map, queue, or tree:

```
a.push (3);
a.push(4);
a.pop(); // returns 4
a.pop(); // returns 3           Circle one: { stack, map, queue, tree }

b.add(3);
b.add(4);
b.next(); // returns 3
b.next(); // returns 4       Circle one: { stack, map, queue, tree }

c.add(100, "OK");
c.add(404, "NOT FOUND");
c.find(100); //returns "OK"
c.exists(299); //returns false Circle one: { stack, map, queue, tree }
```

(c) Briefly describe the purpose of single-step debugging ("SSD"): Your answer should include what a programmer might verify while stepping through code.

(d) An iterator is a class that provides simple sequential access to elements of a data structure. An example iterator is shown below.

```
class ArrayIterator {
    private double[] values;
    private int position=0;

    public ArrayIterator (double [] data) {values = data;}

    public boolean hasNext() {return position<values.length;}

    public double next() { return values[ position ++ ];}
}
```

Complete the main method to create an array iterator for the following array and use your iterator's next and hasNext methods to print out the values of the array.

```
public static void main(String[] args) {
    double[] data = {10.5, 17.5, 31.3, 42.3};

    ArrayIterator it = _____

    while( _____ ) {
        System.out.println( _____ );
    }
}
```