Name: _____

CS 112— Makeup Midterm Exam—Summer 2016

You must do 7 of the 8 problems on the exam. The first and last are mandatory, and you may eliminate any one of problems 2 – 7 by drawing an X through it. Problem 1 is worth 10 points, and all other problems are worth 15. Please write in pen if possible. Circle answers when they occur in the midst of a bunch of calculations. If you need more room, use the back of the sheet and tell me this on the front sheet.

Problem One. (True/False – MANDATORY – 10 points) Write True or False to the left of each statement.

F 1. In Python, narrowing conversions are not allowed.

F 2. A static member of a class should always be made public.

T 3. In a queue implemented as a circular buffer with resizing, a call to dequeue() will never trigger a resize operation.

T 4. The worst case time of insertion sort is the same (in terms of O(....)) as the best case time of selection sort.

F 5. In the boolean expression (C && D) if the expression C evaluates to false, the value of the expression D is returned as the value of the whole expression.

T 6. If an array contains an unordered list of integers, then in general you can not use binary search to find an element.

F 7. In a linked list, a node can never point to itself.

X 8. If a class `Pair` contains a member X and you see an expression `center.X` (and which does not cause an error), then you know that `X` must be declared static and public inside the class `Pair`.

F 9. The type `double` is considered to be a "reference type."

X 10. I have put my name on this exam.

Problem Two (Simulation and Analysis -- 15 points). This has two parts. **(A)** Perform **Insertion Sort** on the following array, according to the algorithm presented in lecture.

3, 9, 8, 4, 6, 1, 0, 2

9 3

9 8 3

9 8 4 3

9 8 6 4 3

9 8 6 4 3 1

9 8 6 4 3 0 1

9 8 6 4 3 2 1 0

Sort so that large elements go to the left and small to the right:

9, 8, 6, 4, 3, 2, 1, 0

6

(B) Answer the following questions, assuming that you are sorting N integers:

What is the best case time of Insertion Sort (in terms of O(....)) and when does it occur?

O(N) WHEN ALREADY SORTED 3

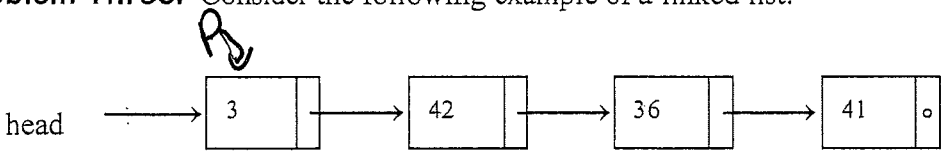
What is the worst-case time (in terms of O(....)) and when does it occur?

$O(N^2)$ REVERSE SORTED 3

What is the average case time (in terms of O(....)).

$O(N^2)$ -3 3

Problem Three. Consider the following example of a linked list:



Apply the following method to this linked list, and show the data structure that would be returned as a result of this method.

SHOW ALL POINTERS

```
Node mystery(Node p) {
  Node q = null;

  for ( ; p != null; p = p.next)
     if (p.item % 2 == 1) {
         q = new Node(p.item, q);
     }
  return q;
}
// Apply the function to the linked list as follows:

Node something = mystery( head );
```

Problem Four. Consider the program shown below.

```
public class ScopeProblem {

    private int a = 4;

    public int someMethod(int b, int c) {
       int d = 4;
       for(int e = 0; e < 10; ++e) {
           if( c < 30 ) {
               String f = "Hi There";
           }
           else {
               System.out.println( _____ );      // first blank
               int g = 8;
           }
        }
        System.out.println( _____ );             // second blank
    }

    public int h = 9;

    private double anotherMethod(int i) {
        while( i < 20 ) {
           double j = 9.8;
        }
        System.out.println( _____ );             // third blank
    }

    private int k = 0;
}
```

(A) Which of the variables a -- k could be used in the first blank?

a, b, c, d, e, h, k

(B) Which of the variables a -- k could be used in the second blank?

a, b, c, d, h, k

(C) Which of the variables a -- k could be used in the third blank?

a, i, h, k

Problem Five. How many times does the following code print out "Hi there!"? Express your answer as a function of N, in terms of O(...N...).

[Hint: Write the O(...N...) estimate for each loop first, and then multiply to get the final result. Partial credit will be given for the estimate for each loop!]

```
for( int i = 0; i < N; i = i + 2 ) {
    for( int k = 1000; k >= 1;  k = k / 2 ) {
        for( int j = 3; j < N; j = j * 2 ) {
            for( int m = 100; m > -N;  --m ) {
                System.out.println("Hi there!");
            }
        }
    }
}
```

$O(N)$

$O(1)$

$O(\log_2 N)$

$O(N)$

$O(N^2 \log_2(N))$

3 items

Problem Six. This problem is about Java and has two parts.

(A) What is the output of the following lines of Java code?

```
int a = 11;
double b = a / 2.0;
int c = (a % 3) / 2;
double d = a * 1000 / 10.0;
double e = (int) ( a * 100.0 / 10 )

System.out.println("a = " + a);

System.out.println("b = " + b);

System.out.println("c = " + c);

System.out.println("d = " + d);

System.out.println("e = " + e);
```

((

5.5

1

1100.0

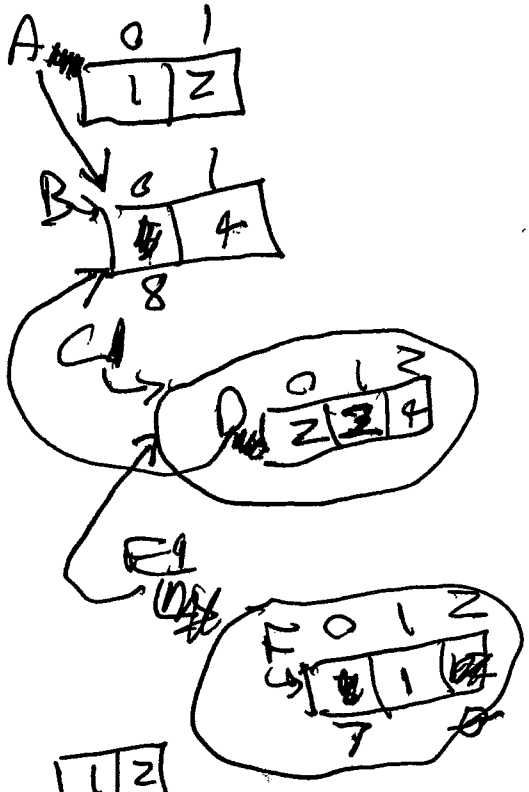
110.0

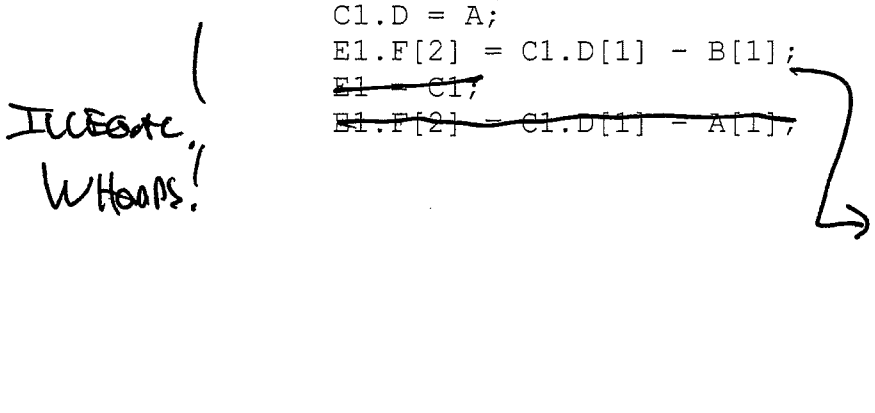
(B) Show what the memory looks like after this code runs (including where all the references A, B, C1, and D1 point to):

```
int[] A = { 1, 2 };
int[] B = { 5, 4 };
class C {
     int[] D = { 2, 3, 4 };
}
C C1 = new C();

class E {
     int[] F = { 3, 1, 12 };
}
E E1 = new E();


C1.D[1] = A[1];
E1.F[0] = C1.D[0] + B[0];
A = B;
A[0] = B[1] + C1.D[2];
C1.D = A;
E1.F[2] = C1.D[1] - B[1];
~~E1 = C1;~~
~~E1.F[2] = C1.D[1] - A[1];~~
```





Illegal, Whoops!

Problem Seven Suppose you have a stack ADT holding integers:

```
class IntStack {

    public void push(int n) ... // etc

    public int pop()  ….        // etc.

    public boolean isEmpty() …  // etc.

    public IntStack() …         // Constructor…

}
```

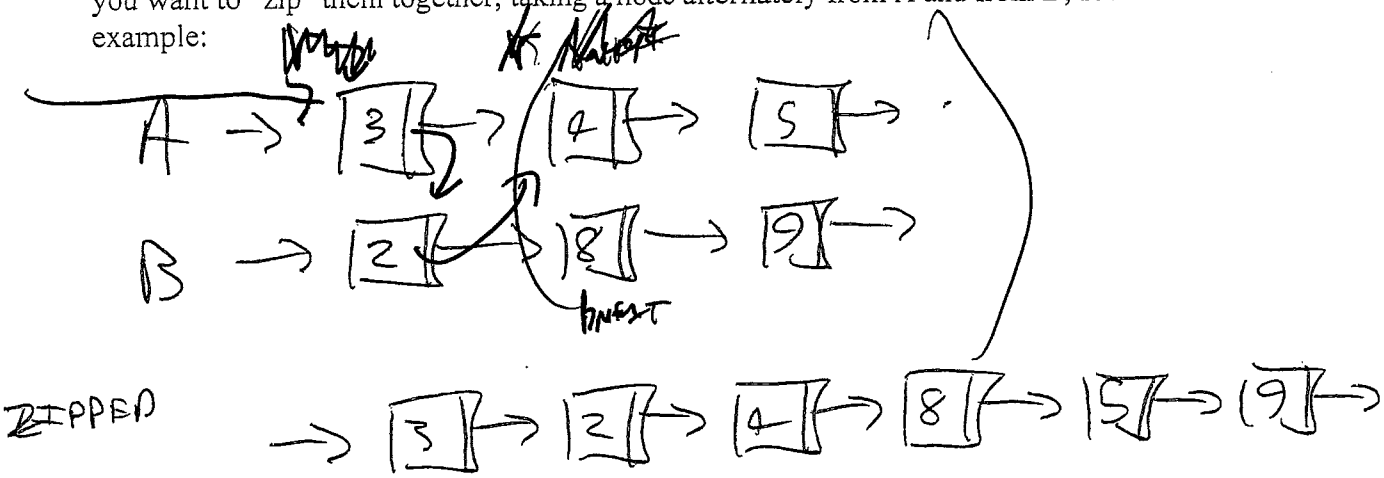
Complete the following template for a method GetMaximum which takes an IntStack and returns the largest integer in the stack; if the stack is empty, it returns the value Integer.MIN_VALUE. Note that at the end of the method, you must have the input stack back in its original order (i.e., you may not destroy the stack). You may not declare an array, but you may declare any other variable, and you may define another stack.

```
int  GetMaximum( IntStack s ) {

    IntStack T = new IntStack();
    INT MAX = INTEGER.MIN_VALUE;
    WHILE (!S.ISEMPTY()) {
        INT TEMP = S.POP()
        IF (TEMP > MAX)
            MAX = TEMP;
        T.PUSH(TEMP);
    }
    WHILE(!T.EMPTY()) {
        S.PUSH(T.POP());
    }
    RETURN MAX;

}
```

Problem Eight. (MANDATORY – 15 points) Suppose you have two lists A and B, and you want to "zip" them together, taking a node alternately from A and from B, for example:



Write a method which returns a list consisting of the zipping together of two lists; you may assume that each of the lists is the same length. You should destroy your lists to do this, that is, do NOT call **new** to create new nodes.

LISTS CAN BE OF ANY LENGTH BUT WILL BE SAME LENGTH.

```
Node zip(Node A, Node B) {
    IF (A == NULL)
        RETURN NULL;
    ELSE{
        NODE R = A.NEXT;
        A.NEXT = B;
        B.NEXT = ZIP(A.NEXT, B.NEXT);
        RETURN R;
    }
}
```

ITERATIVE:

```
NODE ZIP(NODE A, NODE B) {
    NODE R = A;
    WHILE (A != NULL)
        NODE BNEXT = B.NEXT;
        B.NEXT = A.NEXT;
        A.NEXT = B;
        A = B.NEXT;
        B = BNEXT;
        BNEXT = BNEXT + 1
}
```