

Name (PRINT LEGIBLY!) _____ NetID _____
(last, first)

This exam has questions 0, 1, 2, 3, 4, 5 worth a total of 100 points.

Question 0. 2 points. Write your name and netid, legibly, at the top of every page.

Question 1. 18 points. Assume `int` array `m` is sorted (in ascending order). Write a while-loop (not a for-loop) with initialization to perform a binary search for a value `x` in `m`. The postcondition of the loop should be ONE of the following. (You choose which one. The first was used in class; the second, which uses virtual values `m[-1]` and `m[m.length]`, is used in the text.)

R0: $m[0..k] \leq x < m[k+1..m.length-1]$

R1: $m[k] \leq x < m[k+1]$ (assuming $m[-1] = -\infty$, $m[m.length] = \infty$)

You must state the invariant, because you are supposed to develop the loop using it. The invariant and the Java code should be essentially what we did in class or as done in the text. We do not want a complete method, but just the sequence of statements to truthify the postcondition.

Question 2. 20 points. Consider two sorted (in ascending order) **int** arrays **b** and **c**. We want to merge these arrays into a single sorted array **d**. For example, given

$b = \{1, 3, 3, 8\}$ and $c = \{2, 3, 4, 4, 7\}$, **d** is $\{1, 2, 3, 3, 3, 4, 4, 7, 8\}$

Assume these variables are already declared and that the length of **d** is the sum of the lengths of **b** and **c**. Note that an array can have length 0.

Write a single while-loop (with initialization) to merge **b** and **c** into **d**. The precondition is that arrays **b** and **c** are sorted. The postcondition **R** is:

R: d

0	d.length
---	----------

 contains the values of **b** and **c**, sorted

The invariant that you are to use consists of the following three pieces:

	0	h		b.length
b	these values are in d[0..h+k-1]		and	$d[0..h+k-1] \leq b[h..b.length$
- 1]				
	0	k		c.length
c	these values are in d[0..h+k-1]		and	$d[0..h+k-1] \leq c[k..c.length -$
1]				
	0			h+k
d	values of b[0..h-1] and c[0..k-1], sorted			?
d.length				

Write *only one* loop. Don't change the invariant, and don't add extra variables; use only **b**, **c**, **d**, **h**, and **k**. Arrays **b** and **c** should not be changed. Your answer depends entirely on how well you deal with the four loopy questions. A loop that "does the job" but cannot be understood using this invariant gets a grade of 0.

Question 3. 20 points.

On this page is a class Q and a class Computer.

(a) Draw (to the right) one folder of class Q, showing the Object partition of the folder as well as any others.

(b) Execute the call

```
Q.doIt();
```

Write the output of println statements directly beneath the println statements, in the space provided. You need not draw folders or frames for calls for us to see, but drawing at least folders may help you.

```
public class Q {
    public static void doIt() {
        Computer d= new Computer("Mac", 0);
        Computer e= d;
        d.upgradeMemory(64);
        d= new Computer ("Mac", 128);
        Computer f= d;
        f.upgradeMemory(256);
        System.out.println("d: " + d.getMemory() +
            "\n e: " + e.getMemory() +
            "\n f " + f.getMemory());

        Computer c1= new Computer("Windows", 64);
        Computer c2= new Computer("Windows", 128);
        c1.upgradeMemory(64);
        System.out.println("d = e: " + (d == e));

        boolean a= f== d;
        boolean b= c1 == c2;
        System.out.println("a: " + a + "\n b: " + b);
    }
}
```

```
public class Computer {
    private int memory;
    private String type;
    public Computer(String t, int m) {
        type= t;
        memory= m;
    }

    public int getMemory() {
        return memory;
    }

    public void upgradeMemory(int m) {
        memory= memory + m;
    }
}
```

Question 4. 20 points. These questions revolve around the two classes Fahrenheit and Centigrade that appear at the bottom of this page. You may not add any fields to the classes. [Note: Generally, such a class Centigrade would not be a subclass of a class Fahrenheit, because a Centigrade temperature is not a Fahrenheit temperature. Disregard this fact and simply follow the directions, carefully.]

The formulas that relate a Fahrenheit value f and a Centigrade value c are:

$$c = 5*(f - 32)/9, \quad f = 9*c/5 + 32.$$

E.g. 32 degrees Fahrenheit is 0 degrees Centigrade, and 212 degrees Fahrenheit is 100 degrees Centigrade.

- (a) Write the body of the constructor of class Fahrenheit.
- (b) Write the body of the constructor of class Centigrade.
- (c) Finish the return statement in function getCentigrade of class Centigrade.
- (d) Draw an instance (folder) of class Centigrade.
- (e) Indicate which of the following statements are legal:
- Fahrenheit $f1 = \text{new Fahrenheit}(32);$
 - Fahrenheit $f2 = \text{new Centigrade}(6);$
 - String $s1 = f1.toString();$
 - String $s2 = f2.toString();$
 - double $x = f2.getCentigrade();$

```

/** an instance is a temperature in Fahrenheit */
public class Fahrenheit {
    private double f; // the temperature, in Fahrenheit

    /** Constructor: an instance with Fahrenheit
    temperature f */
    public Fahrenheit (double f) {

    }

    /** = this instance's temperature, in Fahrenheit */
    public double getFahrenheit() {
        return f;
    }
}

```

```

/** An instance is a temperature in Centigrade
public class Centigrade extends Fahrenheit {

    /** Constructor: an instance with Centigrade
    temperature c */
    public Centigrade(double c) {

    }

    /** = this instance's temperature, in Centigrade */
    public double getCentigrade() {

        return          ;
    }

    /** = string representation of this temperature */
    public String toString() {
        return "" + getCentigrade();
    }
}

```

Name (PRINT LEGIBLY!) _____

NetID _____

Question 5. 20 points.

(a) Write a single statement that declares an **int** array **b** and initializes it to contain the four values 2, 4, 6, 8.

0 _____ out of 02

1 _____ out of 18

2 _____ out of 20

(b) Write a for-loop that is equivalent to this while-loop (with initialization); don't worry about what the while-loop does:

3 _____ out of 20

```
k= b.length-1;
```

4 _____ out of 20

```
while (k > 0 && b[k] == 5)
    { b[k-1]= k; k= k-2; }
```

5 _____ out of 20

Total _____ out of 100

Name _____

(c) Rewrite the following statement so that it does not use an if-statement or conditional expression.

```
if (atWork || shopping)
    atHome= false;
else
    atHome= true;
```

(d) Below is a method with a loop in its body. Based on our model of execution, describe when each of the variables **p**, **k**, **e**, and **f** are created during execution of a call `m(5)`.

```
public static void m(int p) {
    int k;
    for (k= 0; k != p; k= k+1) {
        int e= k+5;
        System.out.println(e*k);
        int f= e+2;
    }
}
```

Answers 0. David Gries, djg17

```

1. k = -1; h = m.length;
   // inv: m[0..k] <= x < m[h..m.length-1];
   while (h != k+1) {
       int e = (k + h)/2;
       if (m[e] <= x) k = e;
       else h = e;
   }
2. h = 0; k = 0;
   while (h != b.length || k != c.length) {
       if (h != b.length && k == c.length)
           { d[h+k] = b[h]; h = h + 1; }
       else if (h == b.length && k != c.length)
           { d[h+k] = c[k]; k = k + 1; }
       else { // Both b[h], b[k] exist. Put the smaller in d[h+k]
           if (b[h] <= c[k])
               { d[h+k] = b[h]; h = h + 1; }
           else { d[h+k] = c[k]; k = k + 1; }
       }
   }

```

3b. d: 384
 e: 64
 f: 384
 d = e: false
 a: true
 b: false

4. (a) **this.f = f;**
 (b) **super(9*c/5 + 32);**
 (c) **return 5* (getFahrenheit()-32)/9 ;**
 (e) a. Fahrenheit f1 = **new** Fahrenheit(32); legal
 b. Fahrenheit(f2) = **new** Centigrade(6); legal
 c. String s1 = f1.toString(); legal
 d. String s2 = f2.toString(); legal
 e. **double** x = f2.getCentigrade(); illegal

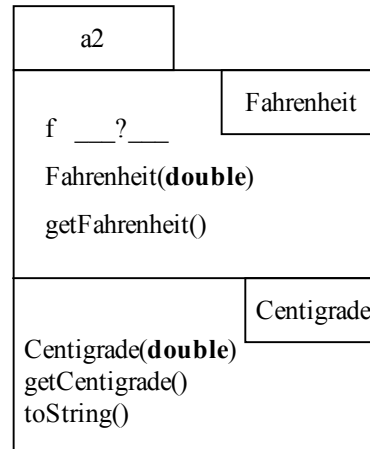
5a. **int[]** b = {2, 4, 6, 8};

5b. **for** (k = b.length-1; k > 0 && b[k] == 5; k = k-2)
 b[k-1] = b[k];

5c. atHome = !(atWork || shopping)

5d. All four variables are created during creation of the frame for the call, before the method body is executed.

4d.



3a.

