ECE 30862 Fall 2015, First Exam

DO NOT START WORKING ON THIS UNTIL TOLD TO DO SO. LEAVE IT ON THE DESK.

THE LAST PAGE IS THE ANSWER SHEET. TEAR IT OFF AND PUT ALL ANSWERS THERE. TURN IN BOTH PARTS OF THE TEST WHEN FINISHED.

You have until 12:20PM to take this exam. The total number of points should be 100. After taking the test turn in both the test and the answer sheet.

Your exam should have 11 (ten) pages total (including this cover page and theanswer sheet, one almost entire blank page and the answer sheet). As soon as the test begins, check that your exam is complete and *let Prof. Midkiff know immediately if it does not.*

This exam is open book, open notes, but absolutely no electronics. If you have a question, please ask for clarification. If the question is not resolved, state on the test whatever assumptions you need to make to answer the question, and answer it under those assumptions. *Check the front board occasionally for corrections.*

I have neither given nor received help during this exam from any other person or electronic source, and I understand that if I have I will be guilty of cheating and will fail the exam and perhaps the course.

Name (must be signed to be graded):

Name (printed, worth 1 pt):

Last four digits of your ID:

For each statement below which has a question number (e.g., **Q7**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put “E” on the answer sheet. **3 points each**.

```
public class B {
  public B( ) {
    System.out.println(“B”);
  }
  private void m1( ) {
    System.out.println(“B”);
  }
  private void m2( ) {
    System.out.println(“B”);
  }
  public void m3( ) {
    System.out.println(“B”);
    m1( );
  }
  public void m4( ) {
    System.out.println(“B”);
    m5( );
  }
  public void m5( ) {
    System.out.println(“B”);
  }
}

public class D extends B {
  public D( ) {
    System.out.println(“D”);
  }
  private void m2( ) {
    System.out.println(“D”);
  }
  public void m3( ) {
    System.out.println(“D”);
    m1( ); // Q1 Answer “OK” or “ERR” for this call
  }
  public void m4( ) {
    System.out.println(“D”);
    m5( ); // Q2 Answer “OK” or “ERR” for this call
  }
  public void m6( ) {
    System.out.println(“D”);
  }
}
```

```
class Main {

  public static void main(String args[]) {

    B b = new B( ); // Q3
    D d = new D( ); // Q4

    b.m1( ); // Q5
    b.m3( ); // Q6
    b.m4( ); // Q7
    b.m5( ); // Q8
    b.m6( ); // Q9

    d.m4( ); // Q10
    d.m5( ); // Q11

    b = d; // Q12 (answer “OK” or “ERR”)

    b.m3( ); // Q13
    b.m6( ); // Q14

    d = b; // Q15 (answer “OK” or “ERR”)


  }
}
```

For each statement below which has a question number (e.g., **Q16**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put “E” on the answer sheet. **3 points each**.

```
public class B {

  public B( ) { }

  public static void m0( ) {
    System.out.println(“B”);
  }

  public static void m1( ) {
    System.out.println(“B”);
  }
  public void m2( ) {
    System.out.println(“B”);
    B.m1( );
    this.m1( ); // ignore any warnings that the compiler
                // and/or Eclipse might give
  }
}
public class D extends B {

  public D( ) { }

  public static void m1( ) {
    System.out.println(“D”);
  }
}

class Main {

  public static void main(String args[]) {

    B b = new B( );
    D d = new D( );

    b.m2( ); // Q16

    D.m0( ); // Q17

    d.m2( ); // Q18

  }
}
```

```
public class B extends Thread {

   static int[ ] a = new int[200];
   static int count = 0;

   int ub, lb;

   public B(int lower, int upper) {
      lb = lower;
      ub = upper;
   }
   public void run( ) {
      for (int i = lb; i < ub; i++) {
         a[count++] = i;
         try {
            sleep(1); // pause 1 ms
         } catch (Exception e) { }
      }
   }

   public void check( ) {
      for (int i = 0; i < count-1; i++) {
         if (a[i] > a[i+1])
            System.out.println("bigger");
      }
   }
}
```

```
class M1 {

   public static void main(String args[]) {
      B t1 = new B(1, 100);
      B t2 = new B(101, 200);

      t1.run( );
      t2.run( );

      try {
         t1.join( );
         t2.join( );
      } catch (Exception e) { }

      t1.check( );

   }
}
```

Q19 true or **false**: The joins make no difference in this program because *t1.join( ),* t2.join( ) *and* t1.check( ) can not execute until *t1.run( )* and *t2.run( )* have both finished. **3 points**.

Q20 true or **false**: *System.out.println(. . .)* will never execute in the *check* function of class *B*. **3 points**.

Q21 true or **false**: This program will use three threads: 1 to execute the main function, 1 to execute the code in *t1.run( )*, and 1 to execute the code in *t2.run( )*. **3 points**.

```
public class B extends Thread {

    static final Object o = new Object( );
    static int count = 0;

    public B( ) { }

    public void run( ) {
        for (int i = 0; i < 100; i++) {
            synchronized(o) {
                int j = count + 1;
                try {
                    sleep(10); // pause for 10 ms
                } catch (Exception e) {}
                count = j;
            }
        }
    }

    public static void print( ) {
        System.out.println(“count = ”+count);
    }
}
```

(a)

```
public class B extends Thread {
    static final Object o = new Object( );
    static int count = 0;
    public B( ) { }
    public void run( ) {
        for (int i = 0; i < 100; i++) {
            int j;
            synchronized(o) {j = count + 1;}
            try {
                sleep(10); // pause for 10 ms
            } catch (Exception e) { }
            synchronized(o) {count = j;}
            System.out.println(count);
        }
    }
}
    public static void print( ) {
        System.out.println(“count = ”+count);
    }
}
```

(b)

```
public class B extends Thread {
    static final Object o = new Object( );
    static int count = 0;
    public B( ) { }
    public void run( ) {
        for (int i = 0; i < 100; i++) {
            int j;
            j = count + 1;
            try {
                sleep(10); // pause for 10 ms
            } catch (Exception e) { }
            count = j;
            System.out.println(count);
        }
    }
}
    public static void print( ) {
        System.out.println(“count = ”+count);
    }
}
```

(c)

```
class M {
   public static void main(String args[]) {
      B t1 = new B( );
      B t2 = new B( );

      t1.start( );
      t2.start( );
      try {
         t1.join( );
         t2.join( );
      } catch (Exception e) { }
      B.print( );
   }
}
```

(d)

The three B classes differ only in synchronization.

Q22 Consider the class *B* in (a) and (b). Which of these will **always** result in *count* having a value of 200 after the program executes, (a), (b), both or neither? Answer on the answer sheet. **5 points**.

Q23 Will the value of *count* printed *B.print( )* in (c) be equal to 200, less than 200, less than or equal to 200, greater than 200 or greater than or equal to 200? You can answer “=”, “<”, “ ≤”, “>” or “≥”. **5 points**.

For each statement below which has a question number (e.g., **Q24**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put “E” on the answer sheet. **4 points each**.

```
public class D {

    public D( ) { }

    public void m1(short s, int i) {
        System.out.println(“Bsi”);
    }

    public void m1(int i, short s) {
        System.out.println(“Bis”);
    }

    public void m1(int i1, int i2) {
        System.out.println(“Bii”);
    }
}
```

```
class Main {

    public static void main(String args[]) {
        short s = 1;
        int i = 1;

        D dO = new D( );

        dO.m1(s, s); // Q24
        dO.m1(i,i); // Q25
        dO.m1((short) i, (int) s); // Q26


    }
}
```

```
public interface I1 {

   public static final int v = 0;

   public void m1( );
   public void m2( );
}

public interface I2 {

   public static final int v = 1;

   public void m2( );
   public void m2(int i);
   public void m3( );
}
```

```
public abstract class A {

   public void m1( ) {
      System.out.println("mi( )");
   }

   public abstract void m4( );
}
public class Main extends A implements I1, I2 {

   public static void main(String args[]) {
      int i = v;
   }
}
```

Answer "Yes" or "No" beside each method if it must be implemented in class Main for the program to compile. 0.5 **points each**.

Q27 public void m1( );

Q28 public void m2( );

Q29 public void m2(int i);

Q30 public void m3( );

Q31 A constructor for I1

Q32 A constructor for I2

Q33 A constructor for A

```
public class E extends Exception {

    String msg;

    public E(String s ) {
        msg = s;
    }

    public String toString( ) {
        return msg;
    }
} public class B {

    public B( ) { }

    public static void m(int i) throws Exception, E {
        if (i < 0) throw new Exception( );
        if (i == 0) throw new E(“E”);
    }
}
```

```
public class Main {

    public static void main(String args[]) {
        for (int i = -1; i < 2; i++) {
            try {
                B.m(i);
            } catch (E e) {System.out.println(“ ”+e+“ ”);
            } catch (Exception e) {System.out.println(“ Ex ”);
            } finally {
                System.out.println(“ ”+i+“ ”);
            }
        }
        System.out.println(“”);
    }
}
```

Q34 Write what is printed by the program above on the answer sheet. **6 points**.

- Ex -1 E 0 1
- Ex E 1
- Ex -1 E 0
- Ex E

```
public class B {

   private int i;

   public B(int v) {
      i = v;
   }

   public void setI(int v) {
      this.i = v;
   }

   public int getI ( ) {
      return i;
   }
}
```

```
public class Main {

   public static void m(B bp1, B bp2) {
      bp1.setI(1);
      bp2.setI(2);
      bp1 = bp2;
   }

   public static void main(String args[]) {
      B b1 = new B(10);
      B b2 = new B(20);

      m(b1, b2);

      System.out.println(b1.getI( )+" "+b2.getI( ));
   }
}
```

Q35, 6 points. Write what is printed by the above program on the answer sheet. It will be:

- 1 2
- 2 1
- 2 2
- 1 1
- 10 20
- 20 10
- 20 20
- 10 10

This page intentionally left almost blank

ECE 30862 Fall 2015 First Exam Answer Sheet

All answers should be on this sheet. Both this sheet and your test must be signed and turned in. You may detach this sheet from the rest of the test to make it easier to write your answers on it. Each question is worth 4 points.

I promise that I have neither Given nor received disallowed aid on this test.

Name (Printed):

Name (Signed):

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

191.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.