Name:_____ NetID:_____

Answer all questions in the space provided. Write clearly and legibly, you will not get credit for illegible or incomprehensible answers. This is a closed book exam. However, each student is allowed to bring one page of notes to the exam. Print your name at the top of every page.

Question:	1	2	3	4	5	6	7	8	Total
Points:	12	12	12	5	10	12	12	15	90
Score:									

1. Multiple choice questions

   (a) What is the name of the method that is run when a Java program executes? (3)

   A. class
   B. main
   C. start
   D. static
   E. void

   (b) If you wanted to store the value of the square root of 2 in a variable, which of the following types would be best? (3)

   A. boolean
   B. char
   C. double
   D. int
   E. String

   (c) Which of the following is *not* a Java keyword? (3)

   A. for
   B. while
   C. do
   D. if
   E. then
   F. else

   (d) Which of the following expressions would evaluate to `1.5`? (3)

   A. `3 / 2`
   B. `(int)3.0 / 2`
   C. `(double)(3 / 2)`
   D. `(double)3 / 2`

2. Why do the following statements not compile? (Explain in one sentence each.)

(a) (3)

```
boolean break = true;
```

(b) (3)

```
System.out.println(Hi there!);
```

(c) (3)

```
if(i = 10) {
    System.out.println(i);
}
```

(d) (3)

```
for(int i = 1 to 10) {
    System.out.println(i);
}
```

3. Given the definitions below, evaluate the following boolean expressions to true or false.

```
boolean apple = true;
boolean orange = false;
boolean banana = true;
boolean kiwi = false;
```

(a) `apple && orange` (2)

(a) _____

(b) `apple || orange || banana` (2)

(b) _____

(c) `!kiwi` (2)

(c) _____

(d) `(apple || kiwi) && (orange || kiwi)` (2)

(d) _____

(e) `!apple && (!orange || banana || kiwi)` (2)

(e) _____

(f) `!apple || (!orange || banana || kiwi)` (2)

(f) _____

4. The following code comes from a solution to the hangman lab. What code is needed on line 15 for the method to work as intended? (5)

```
1  /**
2   * Checks to see if letter occurs at least once in word.
3   * If so, sets the corresponding elements of known to
4   * letter and returns true. If not, returns false.
5   * @param letter Letter that player has guessed.
6   * @param word The word we are checking.
7   * @param known Array of letters that player knows.
8   * @return True if letter was found in word.
9   */
10 public static boolean foundLetter(char letter, String word, char[] known) {
11     boolean found = false;
12     for(int i = 0; i < word.length(); i++) {
13         if(word.charAt(i) == letter) {
14             found = true;
15             // ?????
16         }
17     }
18     return found;
19 }
```

5. As you may recall *even numbers* are numbers that are divisible by 2. The sumEvens method should *return* the sum of all the even numbers in the given array of numbers, returning 0 for an empty array. (10)

```
[1, 2, 3, 4] -> 6
[2, 6, 3, 5] -> 8
[3, 5, 1] -> 0
[2, 2, 2] -> 6
```

```
    public static int sumEvens( int[] numbers ) {
```

6. The following Java program compiles and runs. What is its output? (12)

```
public class ModLoop {

    public static void main(String[] args) {

        int n = 100;
        int z = 2;

        while ( n > 1 ) {
            if( (n % z) == 0 ) {
                System.out.println(z);
                n = n / z;
            } else {
                z = z + 3;
            }
        }
    }

}
```

7. The following Java program compiles and runs. What is its output? (12)

```
public class LoopWithString {

    public static void main(String[] args) {

        String str = "37fun42Java14";
        int n = -1;
        char c = '8';

        while ( !Character.isLetter(c) ) {
            System.out.println("n=" + n + ", c=" + c);
            n++;
            c = str.charAt(n);
        }

        while ( Character.isLetter(c) ) {
            System.out.println("LETTER n=" + n + ", c=" + c);
            n++;
            c = str.charAt(n);
        }
    }
}
```

8. The following Java program compiles and runs. What is its output? (15)

```
public class MethodTest {

    public static int foo(int a) {

        int b = a / 10;
        int c = a % 10;

        System.out.println("a=" + a + ", b=" + b + ", c=" + c);

        for(int i = b; i <= c; i++) {
            System.out.println("i=" + i);
            if(b == c) return i;
        }

        return c;
    }

    public static void main(String[] args) {
        int a = 14;
        int b = 41;
        int c = 33;

        System.out.println("foo(" + a + ")=" + foo(a));
        System.out.println("foo(" + b + ")=" + foo(b));
        System.out.println("foo(" + c + ")=" + foo(c));
    }

}
```