Name:______________________________ UNM Username:______________

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

Question:	1	2	3	4	5	6	7	8	9	10	Total
Points:	10	12	12	10	10	10	8	8	10	10	100
Score:											

1. Consider the following interface representing a counter that starts at zero and counts up by one each time it is incremented. So, it counts 0, 1, 2, 3, 4, . . . (10)

```java
public interface CounterInterface {
    /**
     * Increment counter value by one.
     */
    void inc();

    /**
     * Get the current counter value.
     * @return Current value.
     */
    int getValue();
}
```

Write a simple but complete class named `Counter` that implements this interface. Make sure you use appropriate modifiers on all variables and methods. You may omit javadoc comments.

2. Consider the following classes. What is the output of this code? (12)

```
public class Foo {
    protected int a;
    protected String b;

    public Foo() {
        this("spring");
    }

    public Foo(String a) {
        this(a, a.length());
    }

    public Foo(String a, int b) {
        this.a = b;
        this.b = a;
    }

    public void stuff() {
        System.out.println(a);
        System.out.println(b);
    }

    public void stuff(int a) {
        stuff("CS" + a);
    }

    public void stuff(String b) {
        a /= 2;
        System.out.println(a);
        System.out.println(b);
    }
}
```

```
public class Bar extends Foo {
    protected int c = 2022;

    public Bar(String c) {
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }

    public void stuff(int b) {
        super.stuff("break");
        System.out.println(c);
    }

    public void stuff(String a) {
        stuff();
        System.out.println(a);
    }

    public static void main(String[] args) {
        Foo obj = new Bar("time");
        obj.stuff(251);
    }
}
```

3. Consider the following class. What is the output of this code? (12)

```
public class Baz {
    private String a;
    private static int b;

    public Baz(String a) {
        this.a = a;
        b = a.length();
    }

    public void doThings(String c) {
        System.out.println(a);
        System.out.println(b);
        a = c;
        b += a.length();
    }

    public static void main(String[] args) {
        Baz bob = new Baz("banana");
        bob.doThings("strawberry");

        Baz jane = new Baz("kiwi");
        jane.doThings("pear");

        bob.doThings("peach");
        jane.doThings("pineapple");

        System.out.println(bob.a);
        System.out.println(jane.a);
        System.out.println(b);
    }
}
```

4. Consider the following classes. (10)

```java
public class Parent {

    public void methodA() {
        System.out.println("A");
    }

    public void methodB() {
        System.out.println("B");
    }
}
```

```java
public class Child extends Parent {

    public void methodA() { }
    public void methodB(int n) { }
    public void methodC() { }

    public static void main(String[] args) {

        Parent p = new Parent();
        Parent c1 = new Child();
        Child c2 = new Child();

        // What works here?
    }
}
```

Which of the following lines of code would successfully compile and run without error when placed in the `main` method of `Child` after the code that is already there? Select all that apply.

A. `p.methodA();`
B. `c1.methodA();`
C. `c2.methodA();`
D. `p.methodB();`
E. `c1.methodB();`
F. `c2.methodB();`
G. `Child c = p;`
H. `Child c = c1;`
I. `Child c = c2;`
J. `p.methodB(42);`
K. `c1.methodB(42);`
L. `c2.methodB(42);`
M. `p.methodC();`
N. `((Child)p).methodC();`
O. `c1.methodC();`
P. `((Child)c1).methodC();`
Q. `c2.methodC();`
R. `Object obj = new Parent();`
S. `Parent obj = new Parent();`
T. `Child obj = new Parent();`
U. None of these

5. Consider the following classes.

For each specified method, does the method in `Child` override, overload, or hide the method in `Parent`?

```java
public class Parent {
  protected void method1(int a, String b) {}
  protected static void method2(int i) {}
  public static void method3(int i) {}
  public void method4(int i) {}
  public void method5(Object x) {}
}
```

```java
public class Child extends Parent {
  public void method1(String a, int b) {}
  public static void method2(String s) {}
  public static void method3(int i) {}
  public void method4(int x) {}
  public void method5(String x) {}
}
```

(a) `method1` (2)

(a) ______________________

(b) `method2` (2)

(b) ______________________

(c) `method3` (2)

(c) ______________________

(d) `method4` (2)

(d) ______________________

(e) `method5` (2)

(e) ______________________

6. Select the *single best* answer for each of the following questions.

   (a) Which interface would be the best choice to associate the names of students with their GPA? (2)

   A. `Collection` B. `Deque` C. `List` D. `Map` E. `Set` F. `Queue`

   (b) Which interface would be the best choice as a parameter of a method that can accept any collection of student names, without assuming anything about order or uniqueness? (2)

   A. `Collection` B. `Deque` C. `List` D. `Map` E. `Set` F. `Queue`

   (c) Which type could `foo` be in the following code snippet? (2)

```
int n = foo.size();
```

   A. `Collection` B. `Deque` C. `List` D. `Set` E. `Queue` F. `SortedSet`
   G. Any of these. H. None of these.

   (d) Which type could `foo` be in the following code snippet? (2)

```
Object x = foo.get(0);
```

   A. `Collection` B. `Deque` C. `List` D. `Set` E. `Queue` F. `SortedSet`
   G. Any of these. H. None of these.

   (e) What is the value of the following expression? (2)

   `1 + 2 * 3 + 4 + "5"`

   A. `15` B. `16` C. `"105"` D. `"115"` E. `"135"` F. `"12345"` G. `"1645"`
   H. Some other value. I. The value of this expression is undefined.
   J. This expression would result in a compilation error.

7. Indicate if the following statements are true or false.

   (a) Every exception *must* be caught with a try/catch statement. (1)

   (a) ____________________

   (b) In the expression `3 < 4 || foo(5)`, method `foo` will be called with an argument of 5. (1)

   (b) ____________________

   (c) A class can extend any number of classes. (1)

   (c) ____________________

   (d) A class can implement any number of interfaces. (1)

   (d) ____________________

   (e) Any reference type can be assigned to an `Object` variable. (1)

   (e) ____________________

   (f) It is possible to use an interface as a type. (1)

   (f) ____________________

   (g) An abstract class may contain member variables. (1)

   (g) ____________________

   (h) Every object has a `toString` method. (1)

   (h) ____________________

8. Why don't the following code snippets compile? (Or do they?)

Select the *single* correct answer for each.

(a) (2)

```
public class MyClass {
  private static final int x = 10;

  public static void main(String[] args) {
    x++;
    System.out.println(x);
  }
}
```

A. Variable x is out of scope in the main method.

B. Cannot access private variable x from a public method.

C. Cannot access x without an instance of MyClass.

D. Variable x is a constant, so cannot be incremented in main.

E. Some other error.

F. This code will successfully compile.

(b) (2)

```
List<String> names = new ArrayList<>();
names.put("Jane");
```

A. Cannot assign an ArrayList to a List variable.

B. Missing type parameter on right hand side of assignment.

C. The put method is not part of the List interface.

D. The String `"Jane"` needs to be assigned to a variable.

E. Some other error.

F. This code will successfully compile.

(c) (2)

```
Integer number = 7/2;
```

A. Cannot assign a primitive type to reference type Integer.

B. Cannot assign a double result to an Integer variable.

C. Cannot divide int 7 by int 2.

D. Cannot use "number" as a variable name.

E. Some other error.

F. This code will successfully compile.

(d) (2)

```
Collection<Double> numbers = new Set<>();
```

A. Cannot use Double as generic type parameter.

B. Cannot assign a Set to a Collection variable.

C. Cannot use interface Collection as variable type.

D. Missing type parameter on right hand side of assignment.

E. Set is an interface and cannot be instantiated.

F. Some other error.

G. This code will successfully compile.

9. Write a method that takes a `Collection` of `String` objects and returns the average length of the `Strings` without the character `’x’` in them. (10)

   Any `String` that does contain the character `’x’` should not be included in the mean length calculation. If the collection does not contain any `String` objects without the character `’x’`, return -1.

   Write the method by arranging the following lines of code in the correct order (copying the code, please), adding indentation and closing braces as necessary.

   (a) `public static double averageLength(Collection<String> strs) {`

   (b) `count++;`

   (c) `double sum = 0;`

   (d) `} else {`

   (e) `if (count == 0) {`

   (f) `if (!s.contains("x")) {`

   (g) `for (String s : strs) {`

   (h) `int count = 0;`

   (i) `return sum/count;`

   (j) `return -1;`

   (k) `sum += s.length();`

10. Write a method that takes a `Collection` of `String` objects and returns a `Map` where the keys are initial characters and the values are how many strings starting with that character appears in the input collection. For example, when given a collection containing the Strings “cat”, “dog”, “cougar”, and “Deer” the method should return a map that associates the character ‘c’ with 2, ‘d’ with 1, and ‘D’ with 1 (not necessarily in that particular order, depending on the map used). (10)

    Write the method by copying exactly one line from each of the pairs below, arranging them in the correct order, and adding indentation and closing braces as necessary.

    (a)
    ```
    public static Map<Character, Integer> countChars(Collection<String> strs) {
    public static Map<char, int> countChars(Collection<String> strs) {
    ```

    (b)
    ```
    for(int i = 0; i < strs.size(); i++) {
    for(String s : strs) {
    ```

    (c)
    ```
    Map<char, int> map = new TreeMap<>();
    Map<Character, Integer> map = new HashMap<>();
    ```

    (d)
    ```
    map[c] = n + 1;
    map.put(c, n + 1);
    ```

    (e)
    ```
    if(map.containsKey(c)) {
    if(!map.containsKey(c)) {
    ```

    (f)
    ```
    char c = s.charAt(0);
    char c = s[0];
    ```

    (g)
    ```
    n = map[c];
    n = map.get(c);
    ```

    (h)
    ```
    int n = 0;
    String s = strs.get(i);
    ```

    (i)
    ```
    return map;
    return strs;
    ```