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:	15	15	18	12	10	10	10	10	100
Score:									

1. Multiple choice questions: Select the single correct answer for each.

   (a) Which of the following is *not* a keyword used in exception handling? (3)

   A. finally
   B. catch
   C. final
   D. throw
   E. try
   F. throws

   (b) A static variable with no access modifier could *not* be accessed by: (3)

   A. A static method in the same class.
   B. A non-static method in the same class.
   C. A protected method defined within the same package.
   D. A final method defined in a different package.

   (c) A member declared with a `protected` access modifier is *not always* visible to: (3)

   A. the class in which it is declared.
   B. parent classes of the class in which it is declared.
   C. classes that extend the class in which it is declared.
   D. classes nested inside the class in which it is declared.

   (d) What is the value of the following expression? `"one" + 2 + 3 * 4` (3)

   A. `"one234"`
   B. `"one212"`
   C. `"one14"`
   D. `15`
   E. This expression would result in a compilation error.

   (e) Which interface would be the best choice to hold a collection of unique first names? (3)

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

2. More multiple choice questions: Select the single correct answer for each.

(a) If I want to test if a variable `foo` is an object of type `Bar`, which expression should I use? (3)

A. `Bar instanceof foo`

B. `foo instanceof Bar`

C. `Bar isInstanceOf foo`

D. `foo isInstanceOf Bar`

E. `Bar.isInstanceOf(foo)`

F. `foo.isInstanceof(Bar)`

G. `instanceof(foo,Bar)`

H. `isInstanceOf(foo,Bar)`

(b) Which type could `foo` be in the following code snippet? (3)

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

A. `Collection`

B. `Deque`

C. `List`

D. `Set`

E. `Queue`

(c) What is displayed when the following code is compiled and executed? (3)

```
public class StringCompare {

    public static void main(String[] args) {
        String s1 = new String("Test");
        String s2 = new String("Test");
        if (s1==s2) System.out.println("Same");
        if (s1.equals(s2)) System.out.println("Equals");
    }
}
```

A. `Same`
`Equals`

B. `Equals`

C. `Same`

D. The code compiles, but nothing is displayed upon execution.

E. The code fails to compile.

(d) Which of the following does *not* correctly declare and instantiate a map that associates String keys with Double values? (3)

A. `Map<String, Double> map = new HashMap<String, Double>();`

B. `HashMap<String, Double> map = new HashMap<String, Double>();`

C. `HashMap<String, Double> map = new HashMap<>();`

D. `Map<String, Double> map = HashMap<String, Double>();`

E. `Map<String, Double> map = new TreeMap<String, Double>();`

F. `Map<String, Double> map = new HashMap<>();`

(e) If you wanted to perform custom painting in a `JPanel`, which method would you override? (3)

A. `draw`
B. `drawComponent`
C. `pack`
D. `paint`
E. `paintComponent`
F. `redraw`
G. `repack`
H. `repaint`
I. `repaintComponent`
J. `refresh`

3. Why do the following code snippets not compile? (Explain in one sentence each.)

(a) `List<int> values;` (3)

(b) `Set<String> names = new Set<String>();` (3)

(c) (3)

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

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

(d) (3)

```
public class MyClass {
    public static enum MyEnum {
        YES,
        NO,
        MAYBE;
    }

    public MyEnum enumVal = new MyEnum();
}
```

(e) (3)

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

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

(f) (3)

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

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

```
public class Foo {
    protected double x;
    protected int y;
    protected String z;

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

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

    public Foo(String x, int y) {
        this.x = y + 1.2;
        this.y = y;
        this.z = x;
    }

    public void print(String x) {
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
    }

    public void print(double z) {
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
    }
}
```

```
public class Bar extends Foo {

    public Bar() {
        this("Vacation");
    }

    public Bar(String y) {
        super("Exam");
        System.out.println(y);
        System.out.println(z);
    }

    public void print(int x) {
        print(x / 3);
    }

    public void print(String x) {
        System.out.println(x);
        print(x.length() / 2.0);
    }

    public static void main(String[] args) {
        Foo test = new Bar("Final");
        test.print("CS" + 251);
    }
}
```

5. For this problem, you should write a very simple but complete class that represents a counter that counts 0, 1, 2, 3, 4, . . . (10)

   - The name of the class should be `Counter`.
   - It has one private instance variable representing the value of the counter.
   - It has two public instance methods: `increment()` adds one to the counter value, and `getValue()` returns the current counter value.

   Write a complete definition for the `Counter` class.

6. Write code to create a JButton with text "Click it!" that prints "Click it good!" to the console when it is pressed. Only create the button, you do not have to add it to a layout, show a window, etc. Use an *anonymous class* for the action listener. (10)

7. Consider the following program. What would be displayed when it is run? Draw a picture to illustrate. (10)

```
import java.awt.*;
import javax.swing.*;

public class LayoutExample {

    public static void createAndShowGUI() {
        JFrame frame = new JFrame("Frame Title");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel x = new JPanel();
        x.add(new JButton("A Button"));
        x.add(new JButton("B Button"));

        JPanel y = new JPanel(new GridLayout(3,3));
        for(int i = 1; i < 10; i++) {
            y.add(new JButton("#" + i));
        }

        frame.add(x, BorderLayout.PAGE_END);
        frame.add(y, BorderLayout.CENTER);

        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
                public void run() {
                    createAndShowGUI();
                }
            });
    }
}
```

8. Write a method that takes a `Collection` of `String` objects (any type of collection, not a specific implementation) and returns the length of the longest one. If the collection is empty, return -1. (10)