

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	9	10	Total
Points:	6	18	12	10	6	6	14	12	10	6	100
Score:											

1. Which of the following are *not* Java keywords? Select all that apply. (6)

- A. bool
- B. break
- C. do
- D. enum
- E. import
- F. include
- G. int
- H. switch
- I. if
- J. then
- K. else
- L. elif

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

(a) `Map<String,int> wordCounts;`

(b) `if (x = 5) System.out.println(x);` (3)

(c) _____ (3)

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

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

(d) `boolean break = true;` (3)

(e) `int x = 5` (3)

(f) _____ (3)

```
public class Test {

    private static final int x = 5;

    public void method() {

        for(int i = 0; i < x; i++) {
            System.out.println(i);
        }

        System.out.println(x + ", " + i);
    }
}
```

3. (a) Describe the differences between a checked and an unchecked exception. (4)

(b) Why can't you have a class declared as `final abstract`? (4)

(c) If you have not written a default (parameter-less) constructor for your class, but you have written a constructor that takes at least one parameter, can you still call the default constructor on that class? Why or why not? (4)

4. Multiple choice questions. Select the single best answer.

(a) Which code would you use to instantiate a new `TreeSet` that could *only* hold Strings? (2)

- A. `... = <String>TreeSet();`
- B. `... = new TreeSet<String>;`
- C. `... = TreeSet<String>();`
- D. `... = new TreeSet(String);`
- E. `... = new TreeSet<String>();`
- F. `... = String<TreeSet>();`
- G. `... = new String[TreeSet];`

(b) A member declared with a `protected` access modifier is *not always* visible to: (2)

- 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.

(c) A static variable with no access modifier could *not* be accessed by: (2)

- 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.

(d) What is displayed when the following code is compiled and executed? (2)

```
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("Different");  
        if (s1.equals(s2)) System.out.println("Equals");  
    }  
}
```

- A. Different
Equals
- B. Equals
- C. Different
- D. The code compiles, but nothing is displayed upon execution.
- E. The code fails to compile.

(e) What is the value of the following expression? `2 + 4 + "six" + 8` (2)

- A. "6six8"
- B. "24six8"
- C. "six14"
- D. 20
- E. This expression would result in a compilation error.

5. Write code to create a JButton with text “Click Here” that prints “Button was clicked!” 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. (6)

6. Java uses layout managers to keep track of where components added to a GUI are being displayed. Please pick a layout manager class and describe how components are laid out within it. (6)

7. Remember this mess of code from a quiz earlier this term? (Don't worry, I don't need you to tell me what the output will be this time. Just answer the question on the next page.)

(14)

```
public class Parent {
    public int x = 0;

    public Parent(int x) {
        this.x = x;
    }

    public class InnerChild extends Parent {
        public int x = 1;

        public InnerChild(int x) {
            super(2*x);
            this.x = x;
        }

        public void innerMethod(int x) {
            outerMethod(x);
            System.out.println(x);
            System.out.println(this.x);
            System.out.println(super.x);
            System.out.println(Parent.this.x);
        }
    }

    public void outerMethod(int x) {
        System.out.println(x);
        System.out.println(this.x);
    }

    public static void main(String[] args) {
        int x = 37;
        Parent p = new Parent(x);
        x = 6;
        Parent c1 = p.new InnerChild(x + 1);
        InnerChild c2 = p.new InnerChild(x - 1);

        p.outerMethod(2);
        c1.outerMethod(3);
        c2.innerMethod(4);
    }
}
```

Output:

```
2
37
3
14
4
10
4
5
10
37
```

Which of the following lines of code would cause an error when placed in the main method of Parent after the code that is currently there? Select all that apply.

- A. p.innerMethod(5);
- B. ((InnerChild)p).innerMethod(5);
- C. c1.innerMethod(5);
- D. ((InnerChild)c1).innerMethod(5);
- E. c2.outerMethod(5);
- F. Parent p2 = p;
- G. Parent p2 = c2;
- H. Parent p2 = new Parent(10);
- I. Parent p2 = new Parent(10).new InnerChild(20);
- J. InnerChild c3 = p;
- K. InnerChild c3 = c2;
- L. InnerChild c3 = c2.new InnerChild(20);
- M. InnerChild c3 = new Parent(10);
- N. InnerChild c3 = new Parent(10).new InnerChild(20);

```
CLASS BALL EXTENDS THROWABLE {}
CLASS P{
  P TARGET;
  P(P TARGET) {
    THIS.TARGET = TARGET;
  }
  VOID AIM(BALL BALL) {
    TRY {
      THROW BALL;
    }
    CATCH (BALL B){
      TARGET.AIM(B);
    }
  }
}
PUBLIC STATIC VOID MAIN(STRING[] ARGS) {
  P PARENT = NEW P(NULL);
  P CHILD = NEW P(PARENT);
  PARENT.TARGET = CHILD;
  PARENT.AIM(NEW BALL());
}
}
```

8. Consider the following classes. What is the output of this code?

(12)

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

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

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

    public Foo(String x, int y) {
        this.x = y;
        this.y = y / 8.0;
        this.z = x;
    }

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

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

public class Bar extends Foo {

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

    public void print(int x) {
        print(x * 1.5);
    }

    public void print(String x) {
        print(x.length() / 2);
        System.out.println(x);
        System.out.println(y);
    }

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


9. Write a method that takes a `Collection` of `String` objects and the number of characters in all the `Strings`. In other words, return the sum of the lengths of the `Strings` in the `Collection`. (10)

10. Please organize the following objects in a feasible inheritance hierarchy: *Airplane, Automobile, Blimp, Boat, Car, FlyingMachine, Helicopter, Rowboat, Speedboat, SportsCar, Truck, Vehicle, Yacht* (You may add additional objects if it helps your organization.) (6)