

You have one hour to do this exam.

All programs in this exam must be written in Java.

Name: -----

Id: -----

1. (20 points)

(a) (8 points) Is this a legal Java program? If your answer is yes, what does it print? If your answer is no, explain in two or three sentences what the problem is.

```
class Puzzle {
    public void Move(char c) {
    }
    public static void main(String[] args) {
        Puzzle[] ps = new Puzzle[10];
        ps[0].Move('N');
    }
}
```

- (b) (12 points) Is this a legal Java program? If your answer is yes, what does the program print when executed? If your answer is no, explain in two or three sentences what the problem is.

```
class test {
    public static void main(String[] args){
        int[] A = new int[2];
        A[0] = 3;
        A[1] = A[0];
        A[1] = 4;
        System.out.println(A[0]);

        int[][] B = new int[2][2];
        B[1] = B[0];
        B[1][0] = 78;
        System.out.println(B[0][0]);
    }
}
```

2. (20 points) Scarlett O'Java has written the following program.

```
class main {

    public static void main(String[] args) {
        B ob = new B();
        System.out.println(ob.foo());
        System.out.println(ob.s());
    }
}

class A {
    protected String foo() {
        return "mello " + s();
    }
    private String s () {
        return "yellow";
    }
}

class B extends A {
    protected String s () {
        return "jello";
    }
}
```

- (a) (5 points) Explain in two or three sentences the difference between *overloading* and *overriding* of methods in Java.
- (b) (15 points) What does Scarlett's program print? Explain your answers briefly.

3. (20 points) Write a Java public instance method named `kahuna` that takes an array of integers as input and returns the largest integer in this array. Excessively convoluted code will be penalized.

4. (20 points) Answer the following questions.

- (a) (2 points) How can you introduce subtypes in Java?
- (b) (4 points) What relationship must there be between the type of an object and the type of a reference to that object?
- (c) (3 points) How many supertypes can a Java type have?
- (d) (3 points) Is it possible to change the type of an object by using a `cast`?
- (e) (3 points) What advantage is there in permitting the type of an object to be different from the type of a reference to that object?
- (f) (5 points) Explain the term “dynamic binding” in two or three sentences.

Let `r` be a reference of type `foo`. Consider the invocation `r.do()`. What checks does the compiler perform to make sure this invocation is reasonable? Explain how the method to be invoked (`do`) is found at runtime.