Section 1: Boolean (10 points)

01. F
02. F
03. T
04. F
05. F
06. F
07. F
08. F
09. T
10. F

Section 2: Multiple choice (15 points)

01. (e)
02. (b)
03. (f)
04. (b)
05. (c)

Section 3: Code Reading (15 points)

01. 747
02. 3
03. 4 5 6 7
04. bar foo baz bang
05. 456456
Section 4: Programming (25 points)

public class Die {

    A. (2 points)

    private int numSides;
    private static String color = "red";

    Grading guide:
    Give 1 point for each correct definition of a variable. Must have all the modifiers correct to get the point.

    B. (8 points)

    public Die(int numSides, String color) {
        setNumSides(this.numSides);
        Die.color = color;
    }

    public Die(){
        this(6, Die.color);
    }

    Grading guide:
    2-arg constructor (5 points)
    Give 1 point for the correct signature (param list, name and lack of a return type)
    Give 2 points for setting numSides defensively. (Either check value is non-negative here, or call the setter if it has that logic)
    Give 1 point for usage of the this keyword.
    Give 1 point for setting color statically

    0-arg constructor (3 points)
    Give 1 point for the correct signature (param list, name and lack of a return type)
    Give 2 points for using this() to call the other constructor.

    C. (3 points)

    public void setNumSides(int numSides) {
        if(numSides >= 0) {
            this.numSides = numSides;
        } else {
            this.numSides = 6;
        }
    }

    Grading guide:
    Give 1 point for correct signature (param list, name and return type)
    Give 2 points for correct defensive logic and correct setting of value.
D. (3 points)

```java
public static String getColor() {
    return Die.color;
}
```

**Grading guide:**
- Give 1 point for correct signature (param list, name and return type)
- Give 2 points for making the method static

E. (9 points)

```java
public int rollAgainst(Die that) {
    int one = this.roll();
    int two = that.roll();
    if(one == two) {
        return 0;
    } else if(one > two) {
        return 1;
    } else {
        return -1;
    }
}
```

**Grading guide:**
- Give 2 points for correct signature (param list, name and return type)
- Give 2 points for correctly invoking roll() method on object argument
- Give 2 points for correctly invoking roll() method on current instance (using 'this' is optional)
- Give 2 points for correct logic
- Give 1 point for correct return value

The below code was provided to the students....

```java
public int roll() {
    return 1 + (int)(Math.random() * numSides);
}
```
Section 5: Programming (20 points)

A. (5 points)

```java
public StackOfCharacters(int c) {
    capacity = c < 1 ? DEFAULT_CAPACITY : c;
    list = new Character[capacity];
}
```

Grading guide:
Give 1 point for correctly identifying 'c' as indicating the value for the instance variable capacity
Give 2 points for verifying that the capacity is non-negative, and using default if so. Doesn’t need ternary operator.
Give 2 points for initializing the array properly.

B. (5 points)

```java
public boolean push(Character value) {
    if (!full() && value != null) {
        list[size] = value;
        size++;
        return true;
    } else {
        return false;
    }
}
```

Grading guide:
Give 1 point for checking that the stack is not full
Give 1 point for checking that the value is not null
Give 1 point for putting the value in the array
Give 1 point for updating the size variable
Give 1 point for returning a boolean indicating success or failure of the push operation.

C. (5 points)
public Character pop() {
    if (empty()) {
        return null;
    } else {
        size--;
        return list[size];
    }
}

Grading guide:
Give 1 point for checking that the stack is empty
Give 1 point for returning null if pop() fails for any reason
Give 1 point for updating the size variable
Give 2 points for returning the right element from the array

D. (5 points)

public Character peek() {
    if (empty()) {
        return null;
    } else {
        return list[size - 1];
    }
}

Grading guide:
Give 1 point for checking that the stack is empty
Give 1 point for returning null if peek() fails for any reason
Give 1 point for NOT updating the size variable
Give 2 points for returning the right element from the array