

CS 1713 Exam 2 Solutions — Spring 2012

1) (12 points) a) 'r' b) 3 c) -1 d) 20 e) "rogramming" f) "rogr"

2) (5 points)

```
public String removeLeading(String s) {
    while (s.indexOf(' ') == 0)
        s = s.substring(1);
    return s;
}
```

3) (5 points)

```
public void printReversed(double[] list) {
    if (list.length == 0)
        return;
    System.out.print(list[list.length-1]);
    for (int i=list.length-2; i>=0; i--)
        System.out.print(", "+list[i]);
}
```

4) (5 points)

```
public static int randomRange(int low, int high) {
    Random rand = new Random();
    if (high < low) {
        int temp = high;
        high = low;
        low = temp;
    }
    return low + rand.nextInt(high - low + 1);
}
```

5) (1 point each) a) A, b) D, c) E, d) C, e) B, f) H, g) G, h) I, i) J, j) F

6) (5 points)

```
public boolean isIsocles(Triangle triangle) {
    return (triangle.getSide1() == triangle.getSide2()) ||
           (triangle.getSide1() == triangle.getSide3()) ||
           (triangle.getSide2() == triangle.getSide3());
}
```

10) (8 points)

```
public int throwsUntilN(int n) {
    if ((n < 2) || (n > 12))
        return -1;
    Random rand = new Random();
    int count = 1;
    while (rand.nextInt(6) + rand.nextInt(6) + 2 != n)
        count++;
    return count;
}
```

11) (10 points)

```
public double getWidthOfLargest(Rectangle[] rects) {
    if (rects.length == 0)
        return -1;
    int pos = 0;
    for (int i=1; i<rects.length; i++)
        if (rects[i].getArea() > rects[pos].getArea())
            pos = i;
    // the following 3 lines are for the extra credit
    else if ((rects[i].getArea() == rects[pos].getArea()) &&
            (rects[i].getWidth() > rects[pos].getWidth()))
        pos = i;
    return rects[pos].width;
}
```

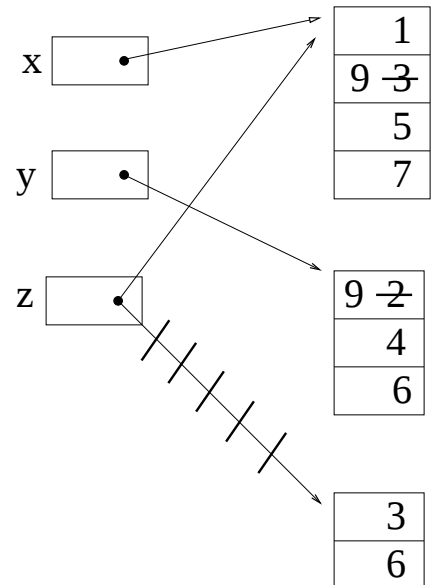
8) (5 points)

x

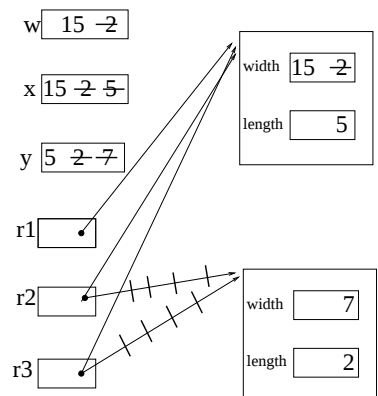
y

z

9) (5 points)



12) (5 points)



7) (25 points)

```
public class Student implements Comparable {

    private String firstName;
    private String lastName;
    private String ssn;
    private String major;
    private int numCredits;

    public Student(String firstName, String lastName, String ssn) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.ssn = ssn;
        major = "undeclared";
        numCredits = 0;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String last) {
        this.lastName = last;
    }

    public String getSsn() {
        return ssn;
    }

    public String getMajor() {
        return major;
    }

    public void setMajor(String major) {
        this.major = major;
    }

    public int getNumCredits() {
        return numCredits;
    }

    public void setNumCredits(int n) {
        numCredits = n;
    }

    public String getFullName() {
        return lastName + ", " + firstName;
    }

    public String toString() {
        return "First name: "+firstName +
            "\nLast name: "+lastName +
            "\nSSN: " + ssn +
            "\nMajor: " + major +
            "\nCredits: " + numCredits;
    }

    public int compareTo(Object other) {
        Student otherStudent = (Student)other;
        int result;
        result = major.compareTo(otherStudent.major);
        if (result == 0)
            result = lastName.compareTo(otherStudent.lastName);
        if (result == 0)
            result = firstName.compareTo(otherStudent.firstName);
        return result;
    }
}
```