

CS 1401, Exam #4, MW 12-1:20 version

Date: Wednesday, November 27, 2013

Name (please type legibly, ideally in block letters):

$$\frac{59}{60} = \frac{98}{100}$$

(4.6)

On November 27, 1895, Alfred Nobel signed his last will and testament in which he used his fortune to set up Nobel Prizes for the best scientific discoveries. In 1895, there was no computer science, so Nobel Prizes are not given to computer scientists:-)

1. Imagine that the Nobel Prize committee decides to award prizes for computing, and you are a member of the Nobel Prize committee deciding on a prize for computing. Describe three people who you would give this prize to, and explain why they are the most deserving.

- 1) Ada Lovelace - Mother of scientific computing, she invented $\frac{10}{10}$ the arithmetic of ~~making~~ making a workable code.
- 2) Steve Jobs - Designed the modern ^{personal} computer. Without his (and his team's) work, the technology we have now would've never been public. $\approx$ exaggerated but good
- 3) The first programmers of Java - While trying to invent a code for home electronics, they developed a language that could be read by any computer, helping the public share programs.

10/10

2a. The more people cite the scientist's work, the bigger the impact of this scientist on the development of science. In precise terms, the impact of a scientist can be measured by the average number of citations of his/her papers. Write a method named citation that, given the total number of citations c and the total number of papers p, returns the average number of citations per paper. In your method, throw an exception to take into account situations when the number of papers is zero.

2b. Throw an additional exception when the number of papers is unrealistically large -- e.g., larger than 10,000.

2c. Call (invoke) your method *citation* in the *main* method to compute the impact of a scientist. Use try-catch in your main method to catch the two exceptions and print the corresponding error messages. On an example when a scientist wrote 10 papers that were cites 120 times, trace, step by step, how the computer will perform the needed computations, and check that the result is indeed correct.

```
public int citation (int c, int p)
{
    if (p > 10,000)
    { throw new UnrealisticOutOfBoundsException(); }
    else if (p == 0)
    { throw new IndexOutOfBoundsException(); }
    else
    { return c/p; }
}
```

$\frac{120}{c}$ $\frac{10}{p}$

$\frac{120}{10}$ 12
c/p

```
public static void main (String [] args)
{ try {
    int avg = citation(120, 10);
    System.out.print(avg);
}
catch (UnrealisticOutOfBoundsException job)
{ S.O.P ("Number of papers in unrealistically large."); }
catch (IndexOutOfBoundsException job)
{ S.O.P ("Not enough papers to calculate."); }
}
```



$\frac{12}{avg}$

prints out
12

10/10

3a. Define a class Scientist whose objects are different programmers. The description of each scientist should contain his/her name, the number of publications, and the number of citations. Your class should contain a constructor method, get- and set-methods, and a method for computing the scientist's impact.

3b. Use your class in the main method to define a new object vonNeumann of type Scientist; assume that he wrote 100 papers and got 2,000 citations. Compute and print his productivity. As the old documents were declassified, it turned out that he had 100 more papers and 1,000 more citations. Use the set-methods to update the values in the corresponding object, and compute and print his updated impact.

3c. Trace your program step-by-step.

```
public class Scientist
{
```

// variables

```
private String name;
```

```
private int p;
```

```
private int c;
```

// constructor

```
public class Scientist (String n, int publi, int cite)
```

```
{ name = n;
```

```
  p = publi;
```

```
  c = cite;
```

```
}
```

// getter methods

```
public String getName()
```

```
{ return name; }
```

```
public int getPublications()
```

```
{ return p; }
```

```
public int getCitations()
```

```
{ return c; }
```

// setter methods

```
public void setName (String newName)
```

```
{ name = newName; }
```

```
public void setPublications (int newPub)
```

```
{ p = newPub; }
```

```
public void setCitations (int newCite)
```

```
{ c = newCite; }
```

// continue Scientist

// impact method

```
public int impact ()
```

```
{ return c/p; }
```

```
} // end class Scientist
```

$\frac{2000}{100} = 20$

$\frac{3000}{200} = 15$

vonNeumann 100 2000
name p c

// main method

```
public static void main (String[] args)
```

```
{
```

```
  Scientist vonNeumann = new Scientist ("vonNeumann", 100, 2000);
```

S.O.P (vonNeumann.getName() + "s productivity was " + vonNeumann.impact());

prints out
vonNeumann's productivity
was 20

// update

```
vonNeumann.setPublications (vonNeumann.getPublications() + 100);
```

```
vonNeumann.setCitations (vonNeumann.getCitations() + 1000);
```

S.O.P (vonNeumann.getName() + " now has an impact of " + vonNeumann.impact());

```
} // end main
```

prints out

vonNeumann now has an impact of 15

black-box testing is not knowing how a machine/code/
equation works, and simply checking if what you get is
accurate to what you inputted.

white-box testing is knowing all the steps and making
sure that every step is accurate

Define Scientist $\{A, B, C\}$ 9/10

4a. Write a method that, given an array of scientists, returns the name of the scientists who has the largest impact.

4b. Test your method in the main program, by applying it to two scientists: vonNeumann and Newton; assume that Newton wrote 50 papers and was cited 10,000 times. Trace the resulting code.

4c. Describe general rules for black-box and white-box program testing.

```

String
public String largestImpact(Scientist[] s)
{
    int biggest = 0;
    for (int i = 0; i < s.length; i++)
    {
        if (s[i].impact() > s[biggest].impact() s[biggest].impact())
        {
            biggest = i;
            biggest = i;
        }
    }
    return s[biggest].getName();
}
// end largestImpact

```

biggest	i	s[i]. impact	s[biggest]. impact	s[biggest]. getName
0	0	15	15	
0	1	10000 50	15	
1		200	15	Newton

~~5000~~

5000

```

// main method
public static void main(String[] args)
{
    // vonNeumann previously defined
    Scientist newton = new new Scientist("Newton", 50, 10000);
    array Scientist j[] = { vonNeumann, newton };
    String bigImpact = largestImpact(j);
    S.OP ("The scientist with the largest impact was " + bigImpact);
}

```

The scientist with the largest impact was Newton

Newton
bigImpact

10/10

sum

5a. Write a method that, given the array *publications* of number of papers written by different faculty from the department, returns the total number of papers published by the department.

5b. In the main method, apply your code to the array consisting of two values 3,000 and 10,000; trace your code, step-by-step, and check that your method returns the correct number $3,000 + 10,000 = 13,000$.

```

public int sumPapers (int p[])
{
    int sum = 0;
    for (int i = 0; i < p.length; i++)
    {
        sum = + p[i];
    }
    return sum;
}

```

```

[0]      [3000]
sum + p[0]
[3000]   [10000]
sum + p[1]
[13000]
sum

```

// main method

```

{
    publications[] = {3000 3000, 10000};
    int H1 = sumPapers (publications);
    S.O.P (H1)
}

```

```

p[0] 3000
p[1] 10000

```

```

[13000]
H1

```

prints out
13000

10/10

6. Give at least two reasons why plagiarism (i.e., copying someone else's work without a proper attribution) is not ethical. For example, is it fair to the author of the original work? Is it fair to the department who hires a person with many publications thinking that he/she has come up with these ideas him/herself? Is it fair to the students who will ask this cheater to supervise their research -- thinking that he/she has a lot of research experience? Give detailed explanations.

Copying someone else's work without citing them is not only rude, but also unfair.

- 1) The person who originally worked on a work ~~had~~ tried their best and worked hard on it. For a person to be lazy and steal someone else's work seems very unfair.
- 2) The plagiarizer might be given a job in the field of what they "wrote" but when they have to do the job they will completely mess it up, as they have no real knowledge in that field.