

CS 1401, Exam #3, TR version

$$\frac{57}{60} \cdot 1.23 = \frac{70}{60} = \frac{117}{100}$$

Date: Thursday, November 20, 2014

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

~~XXXXXXXXXX~~

On November 20, 1985, Microsoft released its Windows 1.0 operating system. You may like Windows you may not, but it is the most widely used operating system.

1a. Write a method named *increase* that, given the total number of computers *c* using Windows in a certain year and the number of computers *f* using it in the following year, returns the percentage with which this number increased.

For those who may have forgotten what a percent is:

- to get a relative increase, you divide the difference $f - c$ by c ;
- then, to get percents, you multiply this ratio by 100.

10
10

1b. In the *main* method, call your method *increase* to compute the percentage increase from 1,000 to 1,500; the result should be 50%.

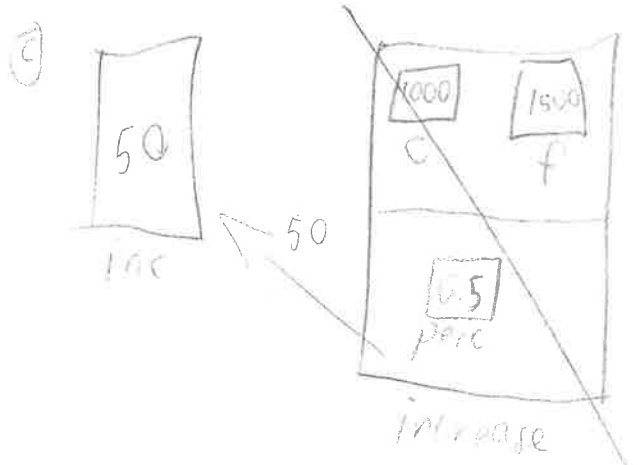
1c. Trace, step-by-step, how the computer will perform all the needed computations, and check that the result is indeed correct.

```

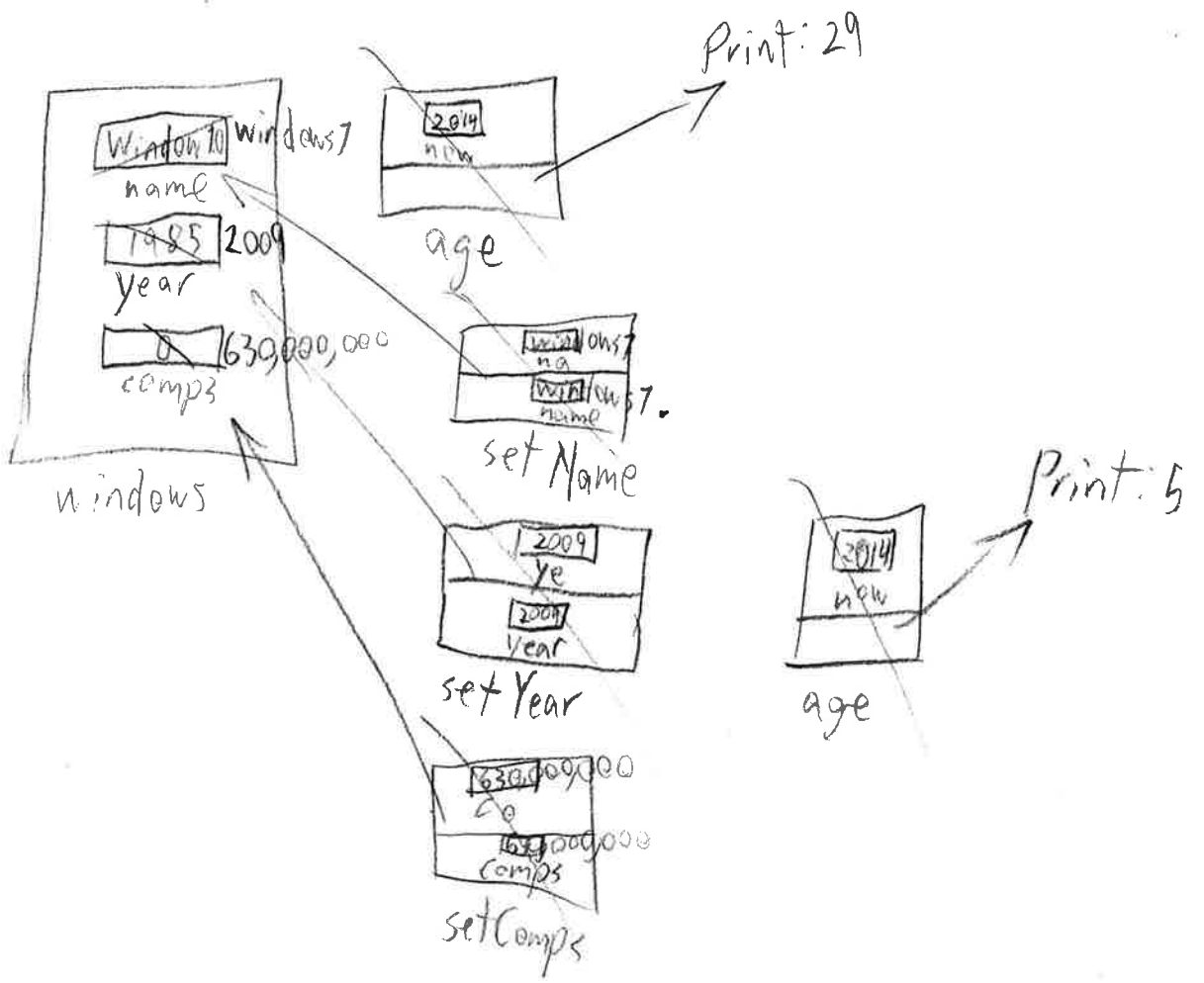
a) public static double increase(int c, int f)
{
    double perc = (f - c) / c;
    return perc * 100;
}
    
```

```

b) in main:
double inc = increase(1000, 1500);
    
```



(C)



2a. Define a class *OperSystem* whose objects are different operating systems. The description of each operating system should contain its name, the year when it was released, and the number of computers using this system. Your class should contain a constructor method, get- and set-methods, and a method that, given a current year, returns the age of the given operating system.

2b. Use your class in the *main* method to define a new object *windows* of type *OperSystem* describing Windows 1.0, which was released in 1985 and now has 0 users. Compute and print the age of this system. Then, change the values corresponding to this object to Windows 7 which was released in 2009 and which now has 630 million users. Compute and print the new age.

2c. Trace your program step-by-step.

a

```

public class OperSystem
{
    private String name;
    private int year;
    private int comps;

    public OperSystem(String n, int y, int c)
    {
        name = n;
        year = y;
        comps = c;
    }

    public String getName() { return name; }
    public int getYear() { return year; }
    public int getComps() { return comps; }

    public void setName(String na) { name = na; }
    public void setYear(int ye) { year = ye; }
    public void setComps(int co) { comps = co; }

    public int age(int now) { return now - year; }
}

```

10
10

b

```

in main.
OperSystem windows = new OperSystem("Windows 1.0", 1985, 0);
SOP(windows.age(2014));
windows.setName("Windows 7");
windows.setYear(2009);
windows.setComps(630,000,000);
SOP(windows.age(2014));

```

Trace is on back
at previous page

3a. Which operating system is the most widely used? To answer this question, write a method that, given an array u of number of users using an operating systems and an array n of their names, returns the name of the most widely used operating system.

3b. To check the correctness of your method, test it on the example of two arrays: array $opSystems$ with elements "Windows 7", "Windows 8", and "Linux", and array $users$ with values 630 million, 40 million, and 100 million.

3c. Trace step-by-step how the piece of code you wrote in Part 3b finds the name of the most widely used operating system.

```

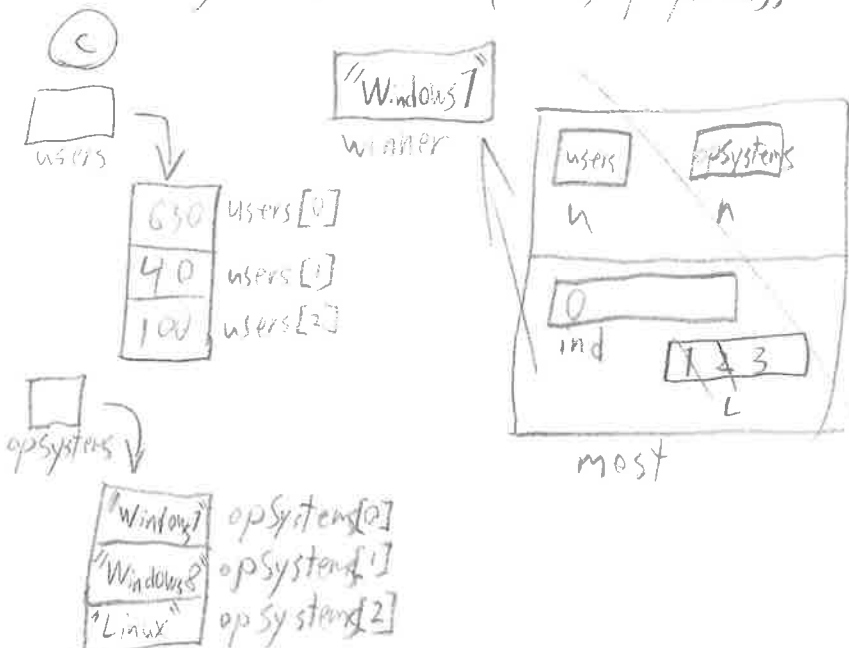
a) public static String most(int[] u, String[] n)
    {
        int ind = 0;
        for (int i = 1; i < u.length; i++)
            if (u[i] > u[ind])
                ind = i;
        return n[ind];
    }
    
```

10
10

b) in main:

```

int[] users = new int { 630, 40, 100 }; // Assume values in millions.
String[] opSystems = new String { "Windows 7", "Windows 8", "Linux" };
String winner = most(users, opSystems);
    
```



4a. Advertisers pay a certain amount for each user. Write a method that, given an array of numbers of users of different operating systems and a payment-per-user value, returns a new array with the amounts given to the owner of each operating system.

4b. To check the correctness of your method, test it in the main program on the example of an array u consisting of 630 million and 40 million users, and a payment of 0.1 cent per user.

4c. Trace step-by-step what happens when you run the piece of code you wrote in Part 4b. The result should be an array consisting of values \$630,000 and \$40,000.

```
public static double[] revenue (int[] users, double fee) {
    double[] earnings = new double [users.length];
    for (int i = 0; i < earnings.length; i++) {
        earnings[i] = users[i] * fee;
    }
    return earnings;
}
```

10
10

```
public static void main (String[] args) {
    int[] u = { 630000000, 40000000 };
    double[] earnings = revenue revenue(u, 0.001);
}
```

① $i = 0$
 $u[0] = 630,000,000$
 $earnings[i] = 630,000,000 * 0.001$
 $630,000,000 * 0.001 = 630,000$
 $earnings[i] = 630,000$

② $i = 1$
 $u[1] = 40,000,000$
 $earnings[i] = 40,000,000 * 0.001$
 $40,000,000 * 0.001 = 40,000$
 $earnings[i] = 40,000$

③ $earnings = \{ 630,000, 40,000 \}$
 return earnings

④ $main\ earnings = method\ earnings$

5. Describe the main teamwork roles, and what are the main tasks of the corresponding persons.

The main roles of teamwork are

- facilitator
- recorder
- time keeper.

10
10

Facilitator:

- should be focused
- should guide the team
- should be transparent and impartial.
- should not allow one person to dominate.
- should make every team member to participate.

Recorder:

- should efficiently record the work being carried by team.
- The recorded data must be clear.

Time keeper: His role is to maintain time management of the team. He should remind the team of its timelines periodically.

10/10

6. Describe what is white-box testing and what is black-box testing. Describe the main rules for testing; give an example for each rule.

white box testing: testing Method that tests the program using the knowledge of program.
 → It ~~also~~ tests all the branches, all the methods involved.

Eg: `public static void main (String [] args) {`

`int x; int a = 2; int b = a + x;`

`if (b >= 3)
return b;`

`else
return a;`

`public int add (int x, int y) {`

`int z = x + y;`

`return z;`

`}`

In white box testing main method is tested separately and method add is also tested separately.

→ white box testing uses all the knowledge used while programming.

Eg:

main		output:
x	0	a
x	1	b
x	2	b
x	-1	a

method add.	output
x y z	
1 2 3	
-1 2 1	
0 1 1	

Black box testing: This is a testing method which targets input and output values without emphasis on knowledge of program steps & methods.

→ It tests the program with simple numbers, border values, and random values.

E.g. public static void main (System args) {

int x; int y;

if (x >= 0)

y = x;

else

y += 2;

}

Simple numbers

x	y	output
0	0	0

-1 1 3

1 1 1

Border value
as its the border for x.
(limit)

Random number

x	y	output
8	8	8