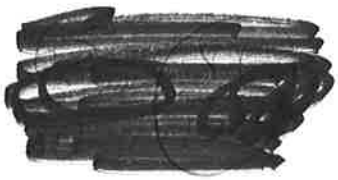


CS 1401, Exam #1, MW version

Date: Monday, September 22, 2014
Name (please type legibly, ideally in block letters):

1. You all probably know that our campus's Bhutanese architecture was inspired by the photos of the Kingdom of Bhutan published in National Geographic magazine. Today is the anniversary of this magazine: its first issue appeared on September 22, 1888.

10/10

- Geographical discoveries became possible when people learned how to calculate their location. Historically the first people to learn these art were ancient Egyptians. Explain what practical needs motivated ancient Egyptians to develop geography-related computations.
- Describe one more event from history of computing.

- Egyptians needed to invent geometry to be able to split the land when the nile River flooded

- Ada lovelace was first female programmer

10/10

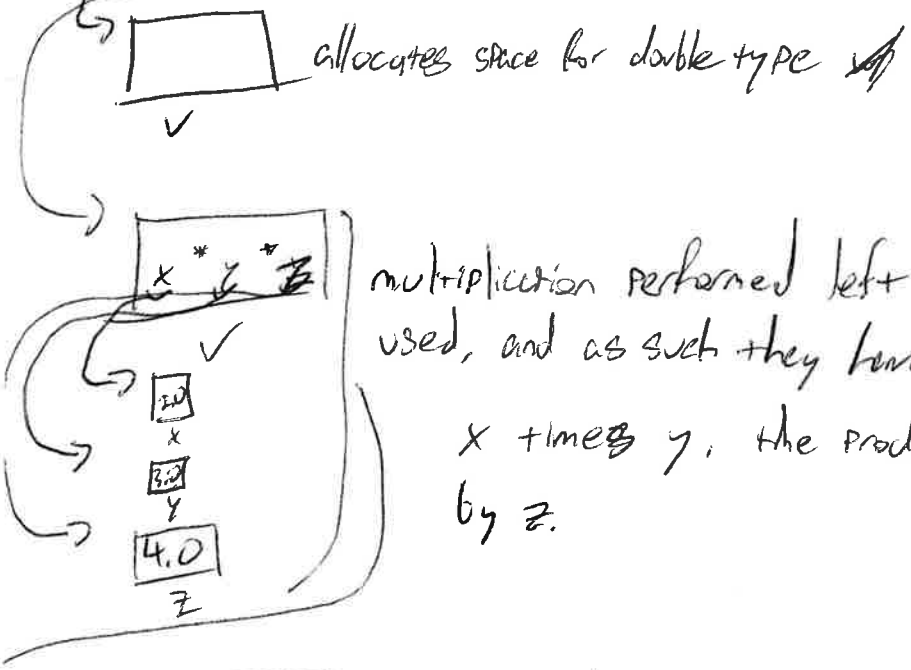
2. For each of the following sequences of symbols, describe which can be valid Java identifiers and which cannot be; if you believe they cannot be, briefly explain why (e.g., "is a reserved word" or "does not start with a letter"):

- NationalGeographic ✓
- main X Keyword
- 1888 X does not begin with a letter
- 22September X see above
- National Geographic X contains a space

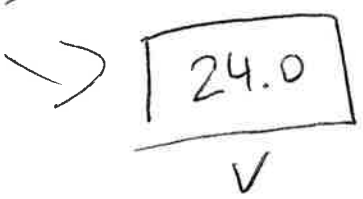
3. The following formula enables us to compute the volume V of a box with sides x, y, and z: V = xyz. Assuming that x, y, and z are already placed in the corresponding variables of type double, write a Java code statement for assigning the corresponding value to the variable v of type double. Explain, step-by-step, which arithmetic operations will be performed first, which next, etc., and trace the computations on the example when x = 2.0, y = 3.0, and z = 4.0. Explain what happens if you simply write xyz in your Java code.

(a) double V = x * y * z;

(b) double V = x * y * z;



multiplication performed left to right as only multiplication is used, and as such they have the same priority. ie, x times y, the product of which is then multiplied by z.



assigns final value to 'v' after reading x, y, and z and multiplying them.

(c) 'xyz' would return an error, as Java would look for a variable of the name 'xyz' and find nothing.

20/20

4-5. Many people subscribe to National Geographic. Let us assume that subscription costs $20 per year. One can pay for several years. Write the main method which asks the user for his/her name, asks for how many years he/she wants to subscribe, and prints a memo describing the price. For example, if Ann Gates wants to subscribe for 4 years, your program should print the following message:

```
From: National Geographic
To: Ann Gates

To subscribe for National Geographic for 4 years, you need to pay
$20 X 4 = $80.
```

Declare 20 as an integer constant, so that it will be easy to change if needed.

Reminder: to read from the keyboard, you can define the reader as follows:

```
Scanner reader = new Scanner(System.in);
```

the header of the *main* method is:

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

~~public class~~

```
Public static void main (String[] args){
    Scanner input = new Scanner(System.in);
    System.out.println("Enter your name.");
    String name = input.next();
    System.out.println("How long would you like to subscribe?");
    int Subyears = input.nextInt();
    final int FEE = 20;
    System.out.println("From: NatGeo");
    System.out.println("to: " + name);
    System.out.println("\nTo subscribe for NatGeo for " + subyears +
            " years, you need to pay");
    System.out.println("$" + FEE + " X " + subyears + " =$" +
            (FEE * subyears) + ".");
```

10/10

6. Suppose that you need to add 3 to the number of years. If the number of years is stored in the integer variable *years*, which of the two lines of code leads to a correct increase:

- years = years + 3.0;
- years = years + 3;

If originally, before each of these two lines, we had 4 years, explain what will happen after each of these lines is implemented by Java. What is a clearer way (different from those above) to add 3 to the variable *years*?

-
```
years = 4;
years = years + 3.0;
// here, the operation of years + 3.0 has a double, so
   all of the operands will be converted to double. Since
   variable years is an integer, you need to cast the
   integer value of the operation because integer type
   is smaller than double type.
```

-
```
years = 4;
years = years + 3;
// Since every value is an integer, the statement will
   add 3 to the current value of years and assign it
   to variable years. years is now 7.
```

- A clearer way would be using augmented expression:

```
years += 3;
```

10/10

7. National Geographic is one of the most popular magazines. Write a piece of code that decides which of the three given magazines is the most popular. The titles of three magazines are stored in the variables *mag1*, *mag2*, and *mag3*, and the numbers of subscribers to each of these magazines is stored in the variables *sub1*, *sub2*, and *sub3*. Use if-then statements to write down a piece of Java code that prints the title of the most popular of the three magazines.

Comment: There is no need to read anything, assume that all six variables have already been assigned values.

```
if (sub1 > sub2 && sub1 > sub3){
    System.out.println(mag1);}
else if (sub2 > sub1 && sub2 > sub3){
    System.out.println(mag2);}
else if (sub3 > sub1 && sub3 > sub2){
    System.out.println(mag3);}
```

what if equal?

8. To pay for the National Geographic subscription, one can either send a check or pay by credit card. Write down a Java statement that uses the known truth values *sentACheck* and *paidByCreditCrad* to assign, to a boolean variable *paid*, true or false depending on whether the subscription was paid or not. Draw the truth tables for "and", "or", and "not". Use these truth tables to find the truth value of your expression when a subscriber sent a check (and did not pay with a credit card).

```
boolean paid = sentACheck || paidByCreditCard;
```

sentACheck	paidByCreditCard	&&	||	!	
true	true	true	true	false	false
true	false	false	(true)	false	true
false	true	false	true	true	false
false	false	false	false	true	true

```
boolean paid = true;
```