Name?

59/60 = (98/100)

B-7

CS 1401, Exam #1, TR version

Date: Thursday, September 19, 2013
Name (please type legibly, ideally in block letters):

---

1. September 19 is International Talk Like a Pirate Day. Nowadays, this is pure fun (and many computer scientists participate in this fun), but in the past, pirates were a serious problems, one of the problems that the British Navy had to deal with. To help the British Navy, Babbage proposed to build a computational device.

- Explain what what was novel in the machine that Charles Babbage proposed to build; did he build it?
- Who helped Charles Babbage with this machine computational task and what was the main contribution of this person?

10+3/10

For extra credit: describe one more event from the history of computing.

a) Charles Baggge proposed to built a machine called an analytical engine, which was capable of performing general ecaclulation through punchcards. This machine was never built.

b) Two people contributed to the conceptualization of the machine. Bagge got the idea of punchcards from the Loom machine built by Jachard and Lady Lovelece (Ader Byron) created several programs for the machine (through punchcards).

- ENIAC Computer: - Fonded by D.O.D.
  - Based on Ferrey digital concept
  - Fully electrical
  - Von Neumen designed its logic

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"):

- Babbage → Valid Identifier
- final → Not Valid: Reserved word to declare a variable as a constant.
- 2013 → Not Valid: Number as the first character.
- 1stTest → Not Valid: Number as the first character
- test-1 → Not Valid: Identifiers can not contain dashes.

8/10

3. To compute the square root of a given number n, Java starts with some approximation a and then computes the more accurate approximation m as follows:

$$m = \frac{1}{2} * \left| a + \frac{n}{a} \right|$$

(For example, when we compute the square root of n = 2.0 and start with a = 1.0, we get a more accurate approximation m = (1/2) * (1.0 + 2.0/1.0) = 1.5.)

Assuming that n and a are already placed in the corresponding variables of type double, write a Java code statement for assigning the corresponding value to the variable m of type double. Explain, step-by-step, which arithmetic operations will be performed first, which next, etc., and trace the computations on the above example. Describe two different ways to avoid getting 0 as the result of evaluating 1/2.

```
double a;
double b;
double m;
```

still 0

```
m = ((a + (n/a)) * ((double) (1/2)));
```

cannot write above the line this on

- or -

but 1.0/2.0

```
m = ((a + n/a)) * ( 1.0 / 2.0 );
```

Order of Operations:

1) Both the division of n/a and 1/2 will be executed

2) The addition of n/a with a (n/a val) will performed

3) The multiplication of (a+(n/a)) * (1/2) will be executed

4) The result will be assign to m.

* To avoid getting a 0, when performing 1/2:

a) Cast 1/2 into a ~~double~~ for the ~~purpose of calculations~~

b) Add a decimal point to both one and two: 1.0 / 2.0

4-5. To successfully defeat a pirate, you need to send twice as many ships as the pirate. Let us write a program which would help create a memo to the Navy describing how many ships we need. Write the main method which asks the user for the name of the pirate, asks how many ships this pirate has, and creates a memo explaining how many ships we need to successfully defeat this pirate. For example, if the pirate's name is Blue Beard and he has 5 ships, your program should print the following message:

```
To: The British Navy
From: Charles Babbage

To successfully defeat Blue Beard, we need to send 10 ships.
```

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){
```

```
import java.util.Scanner;

public class MemoToNavy {

    public static void main(String[] args){
        Scanner scn = new Scanner(System.in);
        String pirateName;
        int numberOfShips;
        System.out.println("Welcome! Please enter the name of the pirate: ");
        pirateName = scn.nextLine();
        System.out.println("Please enter the number of pirate ships:");
        numberOfShips = scn.nextInt();

        System.out.println("To: The British Navy");
        System.out.println("From: Charles Babbage");
        System.out.println("\n" + "to successfully defeat " + pirateName + " " + "ships"
                + ", we need to send " + (numberOfShips * 2));
    }
}
```

10/10

6. Suppose that we need to send one more ship, and, correspondingly, one more captain. If the number of ships is stored in the variable *ships*, and the number of captains in the variable *captains*, which of the two lines of code leads to a correct increase in both:

- captains = ships++;
- captains = ++ships;

If originally, before each line, we had 5 ships and 5 captains, explain what will happen after each of these lines is implemented by Java. What is a clearer way (different from those above) to increase both values?

a) Line #2

b) Line #1:

① The value of ships will get assigned to captains

② Ships will be incremented by one.

Line #2:

① Ships will be incremented by one.

② Captains will get the value of ships (with the addition of 1.)

c)

```
ships = ships + 1;
captains = ships;
```