

600.107: Introduction to Programming Exam 2

Peter H. Fröhlich
phf@cs.jhu.edu

June 19, 2006
Time: 40 Minutes

Start here: Please fill in the following important information using a **permanent pen** before you do **anything** else! Your exam will **not** be graded if you use a pencil or erasable ink on this page.

Name (print): _____

Email (print): _____

Cheat Sheet (yes/no): _____

Ethics Pledge: With your signature you **certify** the information above and you also **affirm** the following:
“I agree to complete this exam without unauthorized assistance from any person, materials, or device.”

Signature: _____

Date: _____

Instructions: Please read these instructions carefully before you start. **Switch off** your phones, pagers, and other noisy gadgets! You are **not** allowed to have anything but a pen (pencil, eraser) and this exam on your desk. You are **not** allowed to talk to anyone during the exam. If you have a question, please raise your hand **quietly**. You must **remain seated quietly** until all exams have been collected. Remember that you can **not** claim grading errors if you do not use a **permanent** pen for your answers.

Do not open before you are told to do so!

You got _____ out of 40 points.

1 Binary Warmup

(10 points)

For each of the following statements, determine whether it is either **true** or **false**. (1 point each)

1. In JAVA, the loop `while (i > 0) { i = i+1; }` will always terminate (i.e. stop running). Assume **i** is an integer.
2. In JAVA, the elements of a record (aka class) must all have the same type.
3. Implicit type conversions are called “casts,” explicit type conversions are called “coercions.”
4. In JAVA, all two-dimensional arrays are square (i.e. number of columns = number of rows).
5. In JAVA, the elements of an array must all have the same type.
6. Assertions express our assumptions about the state of a program at a given point.
7. In JAVA, the “for” and “for-each” loops are completely interchangeable for processing arrays.
8. Unlike in math, variables in imperative programming are entities whose value can change over time.
9. In JAVA, if `a != b` is **true**, then the output of printing **a** and **b** will be different.
10. In JAVA, the brackets “[” and “]” have several different uses.

2 Tough Choices

(8 points)

For each of the following questions, circle **one** answer out of the choices given. (2 points each)

1. In JAVA, which of the following expressions is **equivalent** to `!(a && b)`? (Two expressions are equivalent iff they compute the identical result for all possible values of their constituent expressions.)
 - (a) `(-a) || (-b)`
 - (b) `!a && !b`
 - (c) `a + b == true`
 - (d) `!a || !b`
 - (e) None of the above.
2. In JAVA, which of the following expressions is the **simplest correct** way to check whether the integer `i` is odd? (The expression should evaluate to `true` for **all** odd integers.)
 - (a) `i % 2 == 0`
 - (b) `i % 2 == 1`
 - (c) `i % 2 != 0`
 - (d) `i % 2 == 1 || -1`
 - (e) None of the above.
3. You are developing a JAVA program to compute the grades of students taking a programming course. There are lots of students, each of which has scores for lots of assignments (aside from other information like name and email). Which of the following data structures is **most** appropriate?
 - (a) An array of student records, each of which contains an array of assignment scores.
 - (b) An array of assignments, each of which contains several student records.
 - (c) Two arrays, one for students, one for assignments, and a third to associate them.
 - (d) All of the above work fine, so it doesn't matter.
 - (e) None of the above.
4. In a JAVA program you're reading you find the expression `Math.abs(p - 1.0) < 0.001`. What does it do?
 - (a) It checks whether `p` is even or not.
 - (b) It checks whether `p` is within 0.001 of 1.0 or not.
 - (c) It checks whether `p` is less than 0.0 or not.
 - (d) It checks whether `0.999 < p < 1.001` holds or not.
 - (e) None of the above.

3 Short Answer

(8 points)

For each of the following questions, answer in **one to three** sentences, the shorter the better. (2 points each)

1. A JAVA program contains the instruction `x = y.length + "bla".charAt(z);` which you are **sure** is executed. The program **compiles and runs** without errors. What you can say about the **types and values** that **x**, **y**, and **z** could have? **Explain!**
2. Consider what we called **functions** in JAVA: static methods that return something (i.e. they are not **void**). What is a **side effect** in this context? **Why** can side effects become a problem, especially for functions? **Explain!**
3. KNUTH's proposition: "*Programming is an art, and the job of programmers is to create beautiful programs.*" **Discuss!**
4. Contrast JAVA's **reference types** such as **String** and **arrays** with **value types** such as **int** and **boolean**. **Explain!**

4 Array Reversals

(7 points)

Write a JAVA procedure **reverse**—all of it!—that takes an array of integers as its sole argument and **reverses** that array “in place,” i.e. **without** using a temporary array. Again, you’re supposed to **change** the array given to you, you’re not allowed to make a new one!

5 Adding Arrays

(7 points)

Consider the following JAVA function which is supposed to **add** the two arrays passed to it by **adding their components**. It returns a new array that represents this sum. For example, if we pass the arrays {1, 2, 3, 4} and {0, -1, 1, 6}, we should get a new array with the values {1, 1, 4, 10} back. **Fill in the blanks to make this work!**

```
public static _____ add( int[] a, int[] b )
{
    assert _____;
    int[] c = _____;
    for (int i = 0; _____; i = i + 1)
    {
        c[_____] = a[i] + _____;
    }
    return _____;
}
```

This page is intentionally **mostly** blank in case you run out of space elsewhere. If you ended up here early, please go over **everything** again and remain seated **quietly**! Make sure that the title page is filled out correctly and in **permanent** pen. Maybe you want to "rewrite" your **answers** in permanent pen as well?