

Fall 2003 CS61B Quiz (Midterm) (25/300 points)

;;;

Meta

;;;

GS = Grading Standard

We've tried to include the common errors and grading standard for every question.

;;;;;;

QUESTION 1

;;;;;;

GS: Each question clump tested one idea and was worth 1 full point. This worked out to roughly two blanks per clump. You started the question with 7 full points. Any question wrong in a clump lost the entire clump's points (we write "-1" on your exam). If you missed \*both\* questions in a clump, it was scored the same as if you missed just one. The exceptions to the two-blanks-per-clump were AB (one clump), E (three clumps) and F (three blanks for one clump).

If you had quotes in your output, you lost 1/2 point -- this was only deducted once on the entire page. If you've already lost the point for a clump, we stopped grading (and ignored your extra quotes).

Common errors -- Very few people had every blank correct. Most missed some trivial errors here and there, pretty much spread across every question. Most common error: almost one-third of the class lost clump #7 (f), putting either 10, B or 100 (or all three). About 20 people missed clump #1.

clump #1

- a) 1
- b) one

clump #2

- c) CD
- 61CDEF

clump #3

d) 25  
610

clump #4,5,6 (-1 per wrong answer here)

e) false  
true  
false  
false  
true  
R-ERROR

clump #7

f) 9  
A  
105

QUESTION 2

a) GS: 3 pts  
Start with 3, -1/2 for each incorrect line (min 0)

- (1) AD  
ADE
- (2) BD  
BE  
E
- (3) E
- (4) BE  
B
- (5) E
- (6) C  
CD  
BC  
BCD

(7) @param (optional)

(8)

b) GS: 2 Points

Answer: YES Reasons: E and F

E Because: The Vector class is mutable. If you keep a reference to the vector you pass into FixedVector, you can change your local reference (eg. add a new Account element with a balance greater than that pointed to by maxIndex) and have that change be reflected in the vector reference stored in the FixedVector class.

F Because: Same concept as above, but instead we are changing a single Account object that is stored in the vector in FixedVector.

0 points if you said NO for the first part.

1 point for having E, 1 point for having F.

If you put a correct choice and an incorrect choice, you were deducted 1 point for the incorrect point so your total score would be 0 for this question.

c) GS: 1 Point for both the correct choice and explanation.

Answer is (1) The constructor we are adding has the same signature as an already existing constructor in the class.

We also gave credit if you said that the constructor we are adding has a circular reference with the call to this(2 \* euro) which will prevent it from compiling.

d) GS: 2 Points

Answer is (3) and (6)

You get 1 point for choosing (3) and the correct explanation and you get 1 point for choosing (6) and the correct explanation. If you circled an incorrect choice, 1 point was deducted from your score for this question.

(3) crash only when the input is a Vector with 0 elements (empty)

We gave you a half a point if you said the input is a null Vector

(6) print all but the last element of the input

Some people were confused and thought that the for loop would increment through

2 elements each iteration of the loop, but this isn't the case. Only the call to `e.nextElement()` in the increment part of the for loop is called during each iteration.

.....

### QUESTION 3

.....

```
public int familyFortune() {
    if (myParent == null)
        return(myBalance);
    else
        return(myBalance + myParent.familyFortune());
}
```

```
me.familyFortune());
```

```
public static int familyFortuneStatic(Account a) {
    if (a == null)
        return(0);
    else
        return(a.myBalance + familyFortuneStatic(a.myParent));
}
```

```
Account.familyFortuneStatic(me);
```

### Grading Standards:

Starting with 10 points, scores were determined by deducting up to one point per blank (the two blanks around 'me' in both methods were considered one. Both methods were worth up to five points, even though the second method has six blanks)

\* -1/2 pt for the first instance of a small compile-time error

### Examples:

- Omitting parenthesis in function calls (eg `familyFortune` vs `familyFortune()`... only deducted for the first time).
- Using the wrong variable names (eg `parent` vs. `myParent`).
- Misspelled methods

\* -1 for first instance of severe compile-time errors

Examples:

- familyFortune(myParent) instead of myParent.familyFortune()
- Using Integer instead of int as return type
- Missing Account. in Account.familyFortuneStatic(me)
- Not making the parameter of familyFortuneStatic() an Account

\* -1 for all instances of run-time errors/statements that generate the wrong output.

Examples:

- return(0) instead of return balance().
- Testing myParent.equals(null) (only example of a RT error that only loses a point the first time it was encountered).

\* No points were deducted if the static case does not check that the account passed is null (most people checked that the parent was not null)