

Login name \_\_\_\_\_

**Quiz 2**  
**CSE 131**  
**Winter 2008**

Name \_\_\_\_\_

Signature \_\_\_\_\_

Student ID \_\_\_\_\_

**1. Phase 0 Scoping Fix.** Fill in the blanks of the following Reduced-C program with correct types to test if your fix to the scoping bug present in the starterCode works correctly. If the scoping bug is fixed, this program should compile without error. If the bug is not fixed, this program should generate an assignment error at the line `x = y;`

```
_____ x;          // global x

function : int main() {
    _____ y;

    bool x;          // local x

    x = y;           // If fixed, this line will not cause an error!

    return 0;
}
```

**2. Types, STOs, Modifiable L-vals**

Given the following Reduced-C code fragment:

```
int x = 420;
float y = 4.20;
```

for each statement below state the overall type of the statement, what type of STO you should use to represent this construct, and whether the result is a modifiable l-value (assignable and addressable) or not.

	<u>Resulting Type</u>	<u>Type of STO</u>	<u>Modifiable l-val? (yes/no)</u>
4.2	_____	_____	_____
x	_____	_____	_____
&x	_____	_____	_____
(float *)&x	_____	_____	_____
*(float *)&x	_____	_____	_____
(x + y)	_____	_____	_____

Which Java class / STO from Project I should be the only Java class/ STO that contains a member "m\_value" to hold an actual value?

3. Consider the following record/struct definitions:

A

```
struct fubar {  
    int a;  
    double b;  
    struct fubar c;  
    short d[4];  
};
```

B

```
struct fubar {  
    double a;  
    int b[5];  
    short c;  
    struct fubar d[2];  
};
```

C

```
struct fubar {  
    short a;  
    struct fubar *b;  
    double c[2];  
    int d;  
};
```

Which of the above record/struct definitions is/are semantically correct and why?

4. Semantic Checks/Errors:

Static semantic analysis is done at \_\_\_\_\_ time while dynamic semantic analysis is done at \_\_\_\_\_ time.

Give an example of a static semantic check.

Give an example of a dynamic semantic check (does not have to be related to this Project/Reduced-C).

Give an example of an assignability error.

Give an example of an addressability error.

Typedefs use \_\_\_\_\_ equivalence while structs use \_\_\_\_\_ equivalence for operations such as assignment.