| Name                         | G#   |
|------------------------------|--|
| Group M                      | lember Name:   |
| Group M                      | lember Name:   |
| Today's                      | Goals: We want to get comfortable with symbol resolution in linking.   |
| Work in                      | groups of 2-3 students. Every group will turn in what they've got to Blackboard.   |
| form of a feedback. problems | <b>is based on participation.</b> Get as much done as you can. You will also be given feedback in the 'score' (3-1) and possibly some comments. This doesn't affect your grade – it is solely for . A score of 3 means everything looks great. A score of two indicates some minor s. And a score of one indicates that there were some major issues. If you get a 1, don't panic - go prof or a GTA to get more extensive feedback. |
|                              | <b>Linking Overview</b><br>hat is a symbol in a program?   |
| 2. W                         | hat are the two key functions of a Linker? What does each of these do?   |
| 3. W                         | hich types of symbols are global?  |
| 4. W                         | hat is a <b>static</b> function or <b>static</b> global variable? How are the different than non-static ones?  |

## **Task 2: Symbol Resolution**

1. What is a **Strong** symbol?

## **Task 3: Symbol Resolution in Practice**

```
/* Module A */
int x;
int y = 2;
int main() {
    ...
}
void p1() {
    ...
}
```

```
/* Module B */
double x;
int y;

void p1() {
    ...
}
void p2() {
    ...
}
```

```
/* Module C */
int p2;
char x;
static int y;

void p3() {
...
}
```

For each of the following Modules, circle whether the given symbol is Strong, Weak, or N/A. Then list which Module's version of that symbol will be chosen by the linker.

## **Module A:**

| 1. x is   | Strong | Weak | N/A |
|-----------|--------|------|-----|
| 2. y is   | Strong | Weak | N/A |
| 3. p1 is  | Strong | Weak | N/A |
| Module B: |        |      |     |
| 4. x is   | Strong | Weak | N/A |
| 5. y is   | Strong | Weak | N/A |
| 6. p1 is  | Strong | Weak | N/A |
| 7. p2 is  | Strong | Weak | N/A |
| Module C: |        |      |     |
| 8. x is   | Strong | Weak | N/A |
| 9. y is   | Strong | Weak | N/A |
| 10.p2 is  | Strong | Weak | N/A |
| 11.p3 is  | Strong | Weak | N/A |

Symbol Resolution: Circle which symbol will be picked by references FROM Module A, UNK, or ERROR

| 1. x  | Module A | Module B | Module C | UNK | ERROR |
|-------|----------|----------|----------|-----|-------|
| 2. y  | Module A | Module B | Module C | UNK | ERROR |
| 3. p1 | Module A | Module B | Module C | UNK | ERROR |
| 4. p2 | Module A | Module B | Module C | UNK | ERROR |
| 5. p3 | Module A | Module B | Module C | UNK | ERROR |

## Task 4: Symbol Resolution in Practice

| 1. | In the previous exercise, were there errors for one of the symbols? When do you get an error in symbol resolution?   |
|----|--|
| 2. | In the previous exercise, were there UNK (unknown) for one of the symbols? When do you get an unknown in symbol resolution?  |
| 3. | In the previous exercise, the final part focused on references from Module A. What about references from Module C; which Module's version of y would Module C use? Why is this different from the version of y that Module A uses?                     |
|    | If you still have some time, fire up Zeus and write a two source file program. Try and create these same problems with symbols! See what happens when you have two Strong symbols, two Weak symbols, a Weak and a Strong, and try playing with static! |