

Assignment 2

CMSPC 461: Programming Language Concepts

Prof. Suman Saha, Spring 2023

Due: 11:59 PM, 10th Feb, 2023

Instructions:

You need to submit your homework to **Gradescope**. Do **every problem on a separate page and mark** them before submitting. **If the questions are not marked** or are submitted with **incorrect page-to-question mapping**, the question will **not be graded**. Make sure your name and PSU ID is legible on the first page of your assignment.

We highly recommend you follow the latex/doc template (which can be found on canvas) for the homework submissions, however, if you are writing and scanning make sure you maintain proper handwriting and use stationary of high contrast.

**(Kindly refer syllabus for late submission and academic integration policies.)

Question 1

(4+4 = 8 pt)

Consider the following two codes.

Code A:

```
void globalCounter()
{
    static int count = 0;
    count++ ;
    printf("count= %d \n" , count);
    return 0;
}
int main()
{
    for(int i=0;i<5;i++){
        globalCounter();
    }
    return 0;
}
```

Code B:

```
void globalCounter()
{
    int count = 0;
    printf("count= %d \n" , count);
    count--;
    return 0;
}
```

```

}
int main()
{
    for(int i=0;i<3;i++){
        globalCounter();
    }
    return 0;
}

```

1. How is Code A different from Code B, answer in terms of variable (i.e variable *count*) scope?
2. What will Code A and Code B output as the result?

Question 2

(3+1+1+1+1 = 7 pt)

1. Is C a statically scoped language? Justify your answer with proper reasoning or supporting examples.
2. What is the scope and lifetime of the following in a C function:
 - a) Static Local variable
 - b) Static Global Variable
 - c) Non-Static Local Variable
 - d) Non-Static Global Variable

Question 3

(4 + 2 + 2 = 8 points)

Consider the following pseudo-code. Assume that the language has one global scope, one scope per function and one scope for each braced code block.

```

int x = 10;
int third(int x) {
    print x;
    {
        int x = 30;
        second();
    }
}
int second() {
    print x;
}
int first() {
    int x = 25;
    third(x);
}
first();

```

1. (4 pts) Draw the symbol tables in separation for all the scopes in this program. Assume that each table contains two columns: name and type.

2. (2 pts) If the language uses static scoping rules, what's the expected output from the print statement? Justify your answer by showing the hierarchy of the symbol tables at the print statement.
3. (2 pts) If the language uses dynamic scoping rules, what's the expected output from the print statement? Justify your answer by showing the hierarchy of the symbol tables at the print statement.

Question 4

(6 points)

Consider the following pseudo-code with nested functions. Draw a picture of the run-time stack when the control reaches the end of the last call to f2 (i.e., the control reaches line 5, via the call at the line 7, via the call at line 13). Include dynamic and static links and storage for local variables and parameters. You do not have to show the storage for any other control information or temporaries. But you need to specify execution frames for each function..

```
1   int x = 3;
2   int y = 2;
3   int f1(int b, int c) {
4       int f2(int n) {
5           return n+3;
6       }
7       b = f2(c) + 2*b;
8       return b;
9   }
10
11  int f3(int a) {
12      int b = 5;
13      return f1(b, a);
14  }
15
16  print f3(f1(x,y));
```

Question 5

(5+2+2+2 = 11 points)

Consider the following C++ program:

```
1  int count = 0
2
3  void red() {
4      count += 1
5  }
6
7  void blue(funR) {
8      int count = 0
9      funR()
10     print "blue", count
11 }
12
13 void green() {
14     int count = 0
15     blue(red)
16     print "green", count
17 }
18 }
19
20 void main() {
21     green()
22     print "counterValue", count
23 }
```

What would it print if this language uses:

*Justify your answer by showing the hierarchy of symbol tables at the print statement (assume that each table contains two columns: name and type). (5 points**)*

****(Note: Symbol tables (ST) carry 5 points - we recommend approaching the below questions by creating all the ST and then referring the STs to answer the below questions.)**

1. (2 pts) Static scoping. Justify your answer with a symbol table hierarchy.
2. (2 pts) Dynamic scoping using shallow binding. Justify your answer with a symbol table hierarchy.
3. (2 pts) Dynamic scoping using deep binding. Discuss clearly what happens at line 15, which symbol table is passed?

Question 6

(4+6 = 10 points)

Consider the following class instances in a C++ program:

```
1  static myClass A;
2
3  int main()
4  {
5      myClass *B = foo();
6      ...
7      delete B;
8      return 0;
```

```

9  }
10
11 myClass* foo()
12 {
13     myClass D;
14     myClass* C = new myClass();
15     return C;
16 }

```

- a) (4pt) What is the storage allocation (static/stack/heap) for the *myClass* objects that are reference to by A, C and D? How about the storage for the pointers *B* and *C*?
- b) (6pt) Consider one execution of the program above. The execution trace, a sequence of program statements executed at run time, of this program is

5 13 14 15 6 7 8

For each *myClass* object that is reference to by A, C and D, write down its lifetime (use a subset of execution trace, e.g., 6 7 8 to represent the lifetime).

Optional Question: It wont be graded but is for you to check your understanding. (0 points)

- (a) During the execution of a Java program:
- a) Can a variable be visible but not allocated?
 - b) Can a variable be allocated but not visible?

Explain your answers with proper reasoning.

Suggested reads, which can help you:

- a) Chapter 3 of the text book.
 - b) Brief summary on scope
- (b) In a Java class, a member can have access level modifiers including public, private, protected, and package-private (with no explicit modifier). Describe the difference between those modifiers in terms of the scope of the member being modified.

References **(you may refer any other reference as well):*

<https://docs.oracle.com/javase/tutorial/reflect/class/classModifiers.html>

<https://www.javatpoint.com/access-modifiers>
