Computer Science
Programming Languages Ph.D. Qualifying exam

Spring 2020

Open Book, open note

Start each answer on a new page

Mark each page with problem number and your ID Code.

Problem 1: Scoping and Run-Time

Consider the following pseudo-code in an imperative language:

```
int x = 1, n = 10;

function f () {
   int x = 3;

   function g (int c,d) {
      c = c-1;
      print(c,d,x,n);
    } /* end of g */

 /* body of f */
    g(x,x);
} /* end of f */

function h () {
   int n = 2;

  f();
} /* end of h */

main()  { h(); }
```

A) Suppose that the language uses lexical scoping and pass by value. What is printed?

B) Suppose that the language uses dynamic scoping **and** pass by reference. What is printed?

C) What is the meaning of the word "alias" and how does it relate to part A and/or B?

D) Draw the structure of the stack of activation records at the time when g is active. Your picture should show all the important data that is kept on the stack along with any other memory segments involved with this code.

****************************ANSWER**

A) 2, 3, 3, 10
B) 2, 2, 2, 2
C) Alias means that there are more than one reference to the same memory location. In problem A, there is no case where there is a memory location with 2 references. Pass by value ensures copies are made. In Problem B , pass by reference, any call with a variable makes two pointers to the same memory location. In our case, the call to g() makes 2 additional references to X in the context of G.

s

STACK SEGMENT

G c,d either copies of X from F or pointers to x in F
F x=3
H n=2
Main,

DATA SEGMENT

X = 1
N = 10

CODE SEGMENT – The image of the code in Machine language

HEAP SEGMENT – Not involved

**********END ANSWER **********

Problem 2: Language Implementation

A two-dimensional array may be implemented as a "true" two-dimensional array or as an array of pointers to one-dimensional arrays. Suppose that

- Type "person" is defined as a record (struct) type such that each person record has 50 bytes.
- A[100,75] is an array of person records.
- A is a global variable. Its starting address is known at compile time to be 1000.
- The language uses 0-based indexing.

For each of the two implementations, explain what steps are involved at run-time in computing the address of the expression A[I,J]. (Ignore bounds checking).

Array of pointers:

To access A[i,j], the process requires two steps. First we find the location of A in memory and add i to that location, creating an exact location in memory of the first dimension. We then deference that location, call that V. We then add "j" to V getting is the exact location of "A[i,j]".

For our example, V = memory[1000 + i]. Value = memory[V+ j]
or

Value = memory[memory[1000 + i] + j]

For two dimensional arrays, there is row major and column major (FORTRAN only). We will assume column major. Since Arrays are densely laid out in memory, we need to go far enough into memory to find the correct spot, that is based on the size of the rows (number of columns).

Value = memory[1000 + i*75 + j]

**************************************END ANSWER ***************************

Problem 3: Dynamic Memory

If an object has been allocated on the heap but is no longer accessible, then its space should be reclaimed as free space on the heap.

A) In C++ or Java give **and explain** an example of a small piece of code that creates an object on the heap and then makes the object inaccessible. Your example should be *extremely* small; two lines of code will suffice.

B) In some programming languages reclamation is the responsibility of the programmer. In others such as Java and Scheme it is done automatically by the run time system. What is the technical term for the latter approach?

C) One method for automatic reclamation is to use *reference counts* . Describe briefly how this works (at most 4 or 5 sentences). Under what circumstances does it not work?

**********ANSWER**********

A) JAVA

```
String S;
 S="First";
  S="Second";
```

Java has implicit heap allocation for Strings. Each string is an object on the heap. The "FIRST" object is abandoned when the second assignment occurs.

B) Garbage Collection

C) Reference counting is implemented by adding an additional counting word to each allocation in the heap. The reference count allows multiple variables to all point to the same object. When a variable is reassigned, the reference count goes down. When it hits zero, the memory is now available for reclamation. This does not work with the Object can reference itself directly, of through a sequence of references. A Circularly linked list is an example.

**********END ANSWER**********

Problem 4: Data Types

In Java the type "int" and the type "Integer" are quite different.

A) Give an example of a context in which you can use one but not the other.
B) Explain the difference in their implementation.

A) Anything that requires an object as a parameter. For example, a Generic Stack requires that the elements of the stack be Objects. Same applies for Hash Tables. These Classes require Objects as input as it makes the discovery of the type of Object homogenous. For these classes, the object class Integer would be used

B) an int is 32bits which represents a signed integer. The int can be found in the runtime stack, the data segment and as part of Objects on the heap. An Integer is an Object that resides only on the heap. Objects are implements by having Object identification elements as well as operational elements. In the Integer case, the data in the heap contains a "magic number" to identify the object as an Integer as well as a data element "int" which contains the value. Reference pointers of type Integer would be references into the heap.

****************************END ANSWER ***************************************

Problem 5: Functional Programming

Using your favorite functional language, respond to the following problems. You may only use primitives of the language. It is not a solution to use a built-in function which performs the same action.

Given two indices, I and K, the subsegment is the list containing the elements between the I'th and K'th element of the original list (both limits included). Start counting the elements with 1.

Example in lisp

(subsegment '(a b c d e f g h i k) 3 7)

(C D E F G)

```
(define (subsegment L I K )
  (cond  (( > I 1) (subsegment (cdr L) (- I 1) (- K 1)))
         ((> K 0) (cons (car L) (subsegment (cdr L) I (- K 1))))
         ( else '() )))
```

**********END ANSWER **********