Assignment 2 of CE2004, Principles of Programming Languages

Number of questions: 10

Total Score: 100 points

Due day: 2:00 PM 4th June 2018

P.S.:

(1) You need to type your answers and print them out in papers. And then hand the answer sheets to TAs.

(2) Late submission will not be accepted.

(3) Copying other student's answers is strictly prohibited.

---

(1) (9 points)

What follows is a C program excerpt.

```
void bar()
{  int  tuv[10][10];
        :
        :
}
```

(a) What value does `tuv` represent?

(b) What value does `tuv[1]` represent?

(c) Could we change the value of `tuv[1][1]`?

Ans.

(2) (9 points)

What follows is a Java program excerpt.

```
class xyz{

     :

  void bar()

  {  int  tuv[][]=new int[10][10];

        :

  }

}
```

(a) What kind of values does `tuv` have? (Hine: Using the way described in slide 1-116 of file Ch06.ppt to answer this question.)

(b) What kind of values does `tuv[1]` have? (Hine: Using the way described in slide 1-116 of file Ch06.ppt to answer this question.)

(c) Could we change the value of `tuv[1][1]`?

Ans.

(3) (10 points)

(a) `String s=new String("Welcome");`

Assume the above statement is contained in a Java program and `"Welcome"` does not appear in any other statement in the program, how many `String` objects are created by this Java statement?

(b)
```
String t=new String("Welcome"); // statement 1

String s=new String("Welcome"); // statement 2
```

Assume the above two Java statements are contained in a Java program and statement 1 is right before statement 2, how many `String` objects are created by statement 2?

Ans.

(4) (12 points)

C, C++, and Fortran do not specify range checking of array subscripts. Assume a C compiler allocates memories for arrays declared in the same declaration statement in adjacent areas. For example for the following declaration statement, `int t[10], u[10], v[10];`, the compiler allocates memory for array `t` first. The memory of array `u`'is right after the memory of array `t`. The memory of array `v`'is right after the memory of array `u`. Assume the compiler utilizes row major order to store elements in an array. What follows is a C program.

```
#include<stdio.h>
main()
{ int a[10][10], b[10][10], c[10][10];
  int i,j;

  for(i=0;i<10;i++)
    for(j=0;j<10;j++)
    {
      a[i][j] = 1;
      b[i][j] = 2;
      c[i][j] = 3;
    }
    b[17][9]=4;     /*location 1*/
    b[17][16]=5;    /*location 2*/
    b[-3][7]=6;     /*location 3*/
    b[-3][-6]=7;    /*location 4*/
}
```

(a) The value of which array element will be changed by the statement at location 1?

(b) The value of which array element will be changed by the statement at location 2?

(c) The value of which array element will be changed by the statement at location 3?

(d) The value of which array element will be changed by the statement at location 4?

Ans.

(5) (8 points)

If dynamic scope is used in the following program. at location 10 (a) what is the value of variable a? and (b) what is the value of variable b?

```
int car()
{ int e, f;

  e=7;
  f=8;
  return e+c+a;   //location 1
}
int bar()
{ int c=4, d=5, e=6;
  c=car()+b;  //location 2
  a=5;        //location 3
  e=9;        //location 4
  return c;   //location 5
}
int main()
{ int a, b, c;

  a=1;       //location 6
  b=2;       //location 7
  c=3;       //location 8
  b=bar()+c;  //location 9
  return 1;   //location 10
}
```

Ans.

(6) (12 points) In the following **C** program excerpt,

(a) What is the result printed out by the statement commented as ``Location 6''?

(b) What is the result printed out by the statement commented as ``Location 11''?

(c) What mistake does it make?

```
void bar()
{
   char *p,*q;                /*Location 1*/

   q=malloc(1);               /*Location 2*/
   p=q;                       /*Location 3*/
   *q='h';                    /*Location 4*/
   *p='e';                    /*Location 5*/
   printf("%c",*q);           /*Location 6*/
   q=malloc(1);               /*Location 7*/
   p=q;                       /*Location 8*/
   *q='r';                    /*Location 9*/
   *p='o';                    /*Location 10*/
   printf("%c",*q);           /*Location 11*/
}
```

Ans.

(7) (9 points) In the following **C** program,

(a) Right after the execution of the statement commented as ``Location 1'' is finished, do variable `f` and `i` have the same value?

(b) Right after the execution of the statement commented as ``Location 2'' is finished, do variable `f` and `i` have the same value?

(c) Explain the results of (a) and (b).(P.S.: You can execute this program and observe the results to get your answers.)

```
#include<stdio.h>
main()
{
  float f;
  int   i;

  i=0.07;
  f=i;                          /*Location 1*/
  f=3.57;
  i=f;                          /*Location 2*/
}
```

Ans.

(8) (12 points)

What follows is a C program named `test_address.c`.

```
#include <stdio.h>
int t;
int *p;
main()
{ int q ;

   t=987 ;
   p=&t;
   printf("Hello \n");  /* location 1 */
}
```

Assume after program `test_address.c` is compiled by a C compiler, the addresses of variables `t` and `p` are 678 and 123 respectively. Then after the program is executed and the execution flow arrives at **location 1**, what are the values of `&p`, `p`, and `*p` respectively?

Ans.

(9) (7 points) Consider the following program:

```
procedure Main is
  X, Y, Z : Integer;
  procedure Subl is
     A,Y,Z : Integer;
     procedure Sub2 is
     A,B,Z : Integer;
     begin  -- of Sub2
       . . .
     end;   -- of Sub2
  begin - of Subl
    . . .
  end;  -- of Subl
  procedure Sub3 is
     A,X,W : Integer;
     begin  -- of Sub3
       . . .
     end;   -- of Sub3
begin  -- of Main
 . . .
end;   -- of Main
```

List all the variables, along with the program units where they are declared, that are visible in the bodies of `Subl`, `Sub2`, and `Sub3`, assuming static scoping is used.

Ans.

(10) (12 points) Consider the following program:

```
procedure Main is
   X, Y, Z : Integer;
   procedure Subl is
     A, Y, Z : Integer;
   begin -- of Subl
    . . .
   end;   -- of Subl
   procedure Sub2 is
     A, B, Z : Integer;
   begin -- of Sub2
    . . .
   end;   -- of Sub2
   procedure Sub3 is
     A, X, W : Integer;
   begin -- of Sub3
    . . .
   end;   -- of Sub3
begin  -- of Main
 . . .
end;   --  of Main
```

Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last subprogram activated? Include with each visible variable the name of the unit where it is declared.

(a) `Main` calls `Subl; Subl` calls `Sub2; Sub2` calls `Sub3.`

(b) `Main` calls `Subl; Subl` calls `Sub3.`

(c) `Main` calls `Sub2; Sub2` calls `Sub3; Sub3` calls `Subl.`

(d) `Main` calls `Sub3; Sub3` calls `Subl.`

(e) `Main` calls `Subl; Subl` calls `Sub3; Sub3` calls `Sub2.`

(f) `Main` calls `Sub3; Sub3` calls `Sub2; Sub2` calls `Subl.`

Ans.