

Assignment 1 of CE2004, Principles of Programming Languages

Score: **100** points

Due Time: **14:00 10th April**

P.S.:

(1) You need to type your answers in a file and print them out in answer sheets, then submit your answer sheets to the TAs.

(2) Late submission will not be accepted.

(3) You can discuss these questions with your classmates; however, copying other student's answers is strictly prohibited.

=====

(1) (10 points)

What follows is the content of a C program `example.c`.

```
#include <stdio.h>          /* location 1*/
int bar(int x)              /* location 2*/
{ int a, b, c;              /* location 3*/
  c=x;                       /* location 4*/
  b=x*9;                     /* location 5*/
  a=foo();                   /* location 6*/
  return a;                  /* location 7*/
}                             /* location 8*/
int foo()                   /* location 9*/
{int a=1, b=2, c;           /* location 10*/
  c=a+b;                     /* location 11*/
  return c;                  /* location 12*/
}                             /* location 13*/
int main()                  /* location 14*/
{int a=1, b=2, c=3;         /*location 15*/
  return bar(a);            /* location 16*/
}                             /* location 17*/
```

(a) Are variable `b` defined at location 3, variable `b` defined at location 10, and variable `b` defined at location 15 the same variable?

(b) During run time, at what locations of the above program, variable `b` defined at location 15 and variable `b` defined at location 3 have storage bound to them, but variable `b` defined at location 10 does not have storage bound to it?

Ans.

(2) (6 points)

The CPU of Mary's computer can complete an instruction more quickly than the CPU of Tom's CPU; hence, Mary's computer can always complete a program more quickly than Tom's computer. Is the above statement correct? Give your explanation.

Ans.

(3) (12 points)

What follows is an excerpt of a Javascript program. Assume before location 1, variable `list` has never been used.

```
      :                -- location 1
list = [1, 2]
prefix= list      -- location 2
prefix = 47
list = prefix     -- location 3
      :
```

- (a) At location 1, what is the data type of variable `list`?
- (b) At location 2, what is the data type of variable `prefix`?
- (c) At location 3, what is the data type of variable `list`?

Ans.

(4) (12 points)

A program consists of the following two files, `fileu.c` and `filev.c`

```
/*===== fileu.c =====*/
int a=100;          /* location 1*/
int bar(int y)     /* location 2*/
{int x;            /* location 3*/
  x=y;             /* location 4*/
  return(x);       /*location 5*/
}

/*===== filev.c =====*/
#include<stdio.h>
extern int a;       /*location 6*/
extern int bar(int); /*location 7*/
```

```

main()                                /*location 8*/
{
    printf("a=%d\n", a);
    printf("bar(3)=%d\n", bar(3));
}

```

- (a) List the locations of all variable definitions in the above two files.
 - (b) List the locations of all variable declarations in the above two files.
 - (c) List the locations of all function definitions in the above two files.
 - (d) List the locations of all function declarations in the above two files.
- P.S.: A function formal parameter is also deemed as a variable.

Ans.

(5) (8 points) What follows is a C program.

```

#include <stdio.h>
int total_income, total_visitors_global;

void zoo(char *name, int visitors)
{int adult, children;
    static int total_visitors=0;
        :
    total_visitors=total_visitors+visitors; // location 1
    total_visitors_global=total_visitors;
        :
}

main()
{
    int ticket_price_each_animal_type=2;

    printf("Good Morning!\n"); // location 2
    zoo("giraff", 600);
    zoo("elephant", 300);
    zoo("hippo", 100);
    total_income=ticket_price_each_animal_type*total_visitors_global;
        :
}

```

(a) At location 1, list the names of variables or parameters that have memory assigned to it.

(b) At location 2, list the names of variables or parameters that have memory assigned to it.

Ans:

(6) (10 points)

(a)

```
      :  
a = 3.21;  
      :  
a = "good morning!"  
      :
```

If the above program statements can be successfully executed, which kind of type binding (static or dynamic) is used in the language that is used in the above statements?

(b)

```
      :  
a = 3.21;  
      :  
a = "good morning!";  
      :
```

If the above program statements cannot be transferred to an executable file, which kind of type binding (static or dynamic) is used in the language that is used in the above statements?

Ans.

(7) (6 points) Good language readability can improve writability.

Good language writability is detrimental to readability.

(a) Which one of the above two statements is correct? Which one of the above two statements is wrong?

(b) Give your explanation.

Ans.

(8) (12 points)

```
#include <stdio.h>
int a;
int b=1;
void candy()
{ int c;
  c=100;
}
void bar()
{ int d;
  static int e;

  if(a==3)
    e=b;
  else
    candy();
  a=2;
}
main()
{ int g;

  a=3;      //location 1
  bar();    //location 2
  g=100*b;  //location 3
  bar();    //location 4
  g=200+a;  //location 5
}
```

- (a) For the above program, when the statement at location 1 is executed, how many variables, including static variables and stack-dynamic variables, have been created?
- (b) For the above program, when the statement at location 3 is executed, how many variables, including static variables and stack-dynamic variables, have been created?
- (c) For the above program, when the statement at location 5 is executed, how many variables, including static variables and stack-dynamic variables, have been created?

Ans.

(9) (6 points)

Assume `INTEGER` and `REAL` are special words used to define the data types of variables in a language. Notation ``;` is used to define the end of a statement.

(a) If `INTEGER` and `REAL` are keywords in a language, then are the following statements correct?

```
INTEGER REAL;
REAL INTEGER;
```

(b) If `INTEGER` and `REAL` are reserved words in a language, then are the following statements correct?

```
INTEGER REAL;
REAL INTEGER;
```

Ans.

(10) (8 points) Program optimization can improve programs by making them smaller or faster or both. Hence, when compiling a program, we had better always ask our compiler to optimize our code. Is the above statement always correct? Give your explanation.

Ans.

(11) (10 points)

```
#include <stdio.h>
int a;
int b=1;

void candy()
{ int c;

  c=100;
}

void bar(int p)
{int d;
  static int e;
  :
}
```

```
main()
{ int g;

  bar(3);
  g=100*b;    //location 1
  :
}
```

- (a) For variables a, b, c, d, e, and g in the above program, list the segments (e.g. data segment, stack segment, or BSS segment) that provide storage for the variables in their lifetime.
- (b) There are four variable categories, static, stack-dynamic, explicit heap-dynamic, implicit heap-dynamic. After the above program is executed, at location 1, which categories of variables have been created?

Ans.