

Assignment 1 of CE2004, Principles of Programming Languages

Score: **100** points

Due Time: **24:00 7th April**

(1) Late submission will not be accepted.

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

=====
(1) (6 points)

ML uses a type inference mechanism to determine the type of a variable. What kind (static or dynamic) of type binding is used by **ML** to determine the type of a variable?

Ans.

(2) (12 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.

(3) (9 points)

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

```
      :                -- location 0
a = 100                -- location 1
list = [1, 2]
prefix= list          -- location 2
prefix = 47
list = prefix         -- location 3
      :
```

(a) Right after the statement at location 1 is executed, what is the data type of variable `list`?

(b) Right after the statement at location 2 is executed, what is the data type of variable `prefix`?

(c) Right after the statement at location 3 is executed, 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
extern int t;      // location 2
int bar(int y)     // location 3
{int x;           // location 4
  x=y+t;          // location 5
  return(x);
}                  // location 6

/*===== filev.c =====*/
#include<stdio.h>
int t=9;           // location 7
extern int a;     // location 8
extern int bar(int); // location 9
int main()        // location 10
{ int z;         // location 11
  printf("a=%d\n",a);
  printf("bar(3)=%d\n",bar(3));
}                //location 12
```

- (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.

(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.

(6) (9 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) What are ``special words,’’ ``key words,’’ and ``reserved words?’’

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

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

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

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

Ans.

(7) (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, right after the statement at location 1 is executed, how many variables, including static variables and stack-dynamic variables, have existed?
- (b) For the above program, right after the statement at location 3 is executed, how many variables, including static variables and stack-dynamic variables, have existed?
- (c) For the above program, right after the statement at location 5 is executed, how many variables, including static variables and stack-dynamic variables, have existed?

Ans.

(8) (12 points)

(a) What follows is a C program.

```
# include <stdio.h>
int a;

int bar(int x, int y)
{ int b;

  return b = x+y;
}
int main()
{ int *p;

  p = (int *) malloc (sizeof(int));
  *p = bar (8,9);
}
```

In the above program, (i) which variables are static variables? (ii) which variables are stack dynamic variables? (iii) which variables are explicit-heap dynamic variables?

P.S.: A function formal parameter is also deemed as a variable.

(b) What follows is a C++ program.

```
using namespace std;
class Person {
public:
  string name;
  string gender;
  int age;
  Person(string x, string y, int z) {
    name = x; gender = y; age = z;
  }
  Person() {
    name = "Mary"; gender = "Female"; age = 20;
  }
};
void bar()
{ static int w=0;
  ++w;
}
```

```

int main() {
    int i;
    Person peopleA;
    Person *peopleB;

    peopleB = new Person(); //location 1
    bar();
}

```

- (i) In the above program, which variables are static variables?
- (ii) Right after the statement at location 1 is executed, how many variables with type class Person exist?
- (iii) Are variables in the above question, i.e. question (8)-(b)-(ii), allocated from the same area (stack, heap, BSS, DATA)?

Ans.

(9) (8 points)

During the lifetime of a variable, the variable must be bound to a memory area. (1) Will the memory area bound to a variable always be the same? (2) Give your explanation.

Ans.

(10) (6 points)

What follows is a **Fortran 90** program.

```

PROGRAM Hello
  IMPLICIT NONE
  INTEGER :: Patrick island, a
  PRINT *, 'Welcome to Fortran'
  a=6
  Pat rick is land = 19
  Patrick island = Pat rick is land + a      ! line 8
END PROGRAM Hello

```

After line 8 of the above program is executed, (i) what is the value of variable Patrick island? (ii) what is the value of variable Pat rick is land?

Ans:

(11) (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.