# Programming Design, Spring 2015
# Midterm Exam

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

**Note.** You do not need to return these problem sheets. Write down all your answers on the answer sheets provided to you.

1. (20 points; 2 points each) For the following statements, Write down "T" if one is true or "F" if it is false. DO NOT provide any explanation.

   (a) C++ is a high-level object-oriented programming language.

   (b) A for loop can always be replaced by a while loop.

   (c) A while loop can always be replaced by a for loop.

   (d) A local variable may hide a global variable. When this happens, it is possible that one may not access that global variable.

   (e) A local variable may hide another local variable. When this happens, it is possible that one may not access that hided local variable.

   (f) In a global function, one may access a member variable of a structure.

   (g) In a member function of a structure, one may access a global variable.

   (h) If we execute the statement `int a[10][10];` to declare an array `a`, `&a[0][5] - a[0]` and `&a[0][5] - a[0][0]` will both be 5.

   (i) If we execute the statement `int a[10][10];` to declare an array `a`, `&a[0][5] - a[0]` and `&a[0][7] - (a[0] + 2)` will both be 5.

   (j) A constructor cannot have a default argument. This is why we need to overload constructors.

2. (20 points; 5 points each) Briefly answer the following questions. Your answers need to be precise but need not to be long.

   (a) Explain what is the dangling problem and how to avoid it.

   (b) Explain what is calling by reference and when to use it.

   (c) Write down one reason to use a constant variable.

   (d) Write down at least two benefits of using a class rather than a structure.

3. (10 points) You are given an $n \times n$ matrix $A$. Your task is to find for how many pairs of $i$ and $j$ we have $A_{ij} \neq A_{ji}$, $i = 1, ..., n$, $j = 1, ..., n$. For example, if

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 6 & 7 & 8 & 9 \end{bmatrix},$$

   the answer is 3: $A_{14} \neq A_{41}$, $A_{24} \neq A_{42}$, and $A_{34} \neq A_{43}$. Suppose that the matrix $A$ is given as a two-dimensional C++ integer array `mat`. `mat` has `n` rows, each having `n` elements. Write a few lines of C++ codes to find the answer. You may assume that `mat` and `n` are given in a proper way, `n` is positive and below 1000, and all elements in `mat` are within $-1000$ and 1000.

4. (30 points; 5 points each) Consider the following structure (please assume that `<cmath>` has been included in the program defining this structure):

```
struct Randomizer
{
  int cur[2];
  int rand();
  bool srand(unsigned int s);
};
int Randomizer::rand()
{
  int curNum = cur[0] * 10 + cur[1];
  curNum = curNum * curNum;
  int next[4] = {0};
  for(int i = 3; i >= 0; i--)
  {
    int digit = floor(curNum / pow(10, i));
    next[3 - i] = digit;
    curNum -= digit * pow(10, i);
  }
  cur[0] = next[1];
  cur[1] = next[2];
  return cur[0] * 10 + cur[1];
}
bool Randomizer::srand(unsigned int s)
{
  if(s < 100)
  {
    cur[0] = floor(s / 10);
    cur[1] = s % 10;
    return true;
  }
  else
    return false;
}
```

Answer the following questions independently.

(a) Write down the output of the following program:

```
int main()
{
  Randomizer r;
  r.srand(36);
  for(int i = 0; i < 10; i++)
    cout << r.rand() << " ";
  return 0;
}
```

(b) The structure uses a specific algorithm to produce pseudorandom numbers. Use your own words to explain how this algorithm works. Is it a good algorithm? Why or why not?

(c) Should we check whether `s` is negative in the member function `srand()`? Why or why not?

(d) Consider the member function `rand()`. Modify the `for` loop to an equivalent `while` loop.

(e) In the current implementation, each `Randomizer` variable contains two integers. Propose a way to cut down the amount of memory required for each `Randomizer` variable. Precisely explain what should be changed for member variables; briefly explain (in words) what should be changed for member functions. If you believe that cutting down memory usage is impossible, explain why.

(f) Suppose you want to modify this structure into a class. Determine whether you need a constructor, copy constructor, and/or destructor for the new class. Explain why.

5. (20 points; 5 points each) We need to write a program to record the relationship for a group of people. In our program, each person is stored as an object of the class `Person`:

```
class Person
{
private:
  const int id;
  int n;
  int* m;
public:
  Person(int id);
  Person(int id, int n, int* m);
  ~Person();
  void knowPerson(int anId);
  void print();
};
```

Each person has a unique ID (in `id`), the number of people she knows (in `n`), and the list of people that she knows (in the dynamic array `m`). Besides the constructors and destructor, the function `knowPerson` expand the list of known people by one and add `anId` into the list, and the function `print` prints out the list of known people. The functions are implemented below:

```
Person::Person(int id)
{
  this->id = id;
  n = 0;
  m = NULL;
}
Person::Person(int id, int n, int* m)
{
  this->id = id;
  this->n = n;
  this->m = m;
}
Person::~Person()
{
  delete m;
}
void Person::knowPerson(int anId)
{
  n++;
  int* temp = new int[n];
  for(int i = 0; i < n - 1; i++)
    temp[i] = m[i];
  temp[n - 1] = anId;
  m = temp;
  delete [] temp;
}
void Person::print()
{
  for(int i = 0; i < n; i++)
    cout << m[i] << " ";
  cout << "\n";
}
```

(a) Correct the errors (one or many) in the constructors and destructor. If you believe that there is no error, write down "No error."

(b) Correct the errors (one or many) in `knowPerson`. If you believe that there is no error, write down "No error." You may assume that `anId` is not the same as any ID in `m`.

(c) Suppose that all errors in the constructors, destructor, and `knowPerson` are corrected. Now we add a global function `globalPrint` and use it in the main function in the following way:

```
void globalPrint(const Person& p)
{
  p.print();
}
int main()
{
  Person p(1);
  p.knowPerson(2);
  p.knowPerson(4);
  globalPrint(p);
  return 0;
}
```

We will see a compilation error. Explain why and how to modify the class to fix it.

(d) Ignore Part (c). Suppose that all errors in the constructors, destructor, and `knowPerson` are corrected. Now we add a global function `globalPrint2` and use it in the main function in the following way:

```
void globalPrint2(Person p)
{
  p.print();
}
int main()
{
  Person p(1);
  p.knowPerson(2);
  p.knowPerson(4);
  globalPrint2(p);
  return 0;
}
```

We will see a run time error. Explain why and how to modify the class to fix it.