CS100 Final Exam Cover Sheet

Instructors: Laurent Kneip, Jie Zheng, Fu Song
January 16, 2019

PRINT your name: (last name)________________________, (first name)__________________

PRINT your email ID: ________________________________@shanghaitech.edu.cn

PRINT your student ID:________________________________

PRINT your tutorial group number:______________________

INSTRUCTIONS:

- You have 110 minutes (~09:10-11:00) to complete the exam.
- Your exam will not be graded unless you complete the above section and the cover sheet, and turn in both this exam book and the cover sheet.
- This exam is **closed-book and closed-notes**, and **no electronic devices are permitted**.
- Mark your answers on the exam itself. We will not grade answers written on scratch paper.
- Your performance is supposed to reflect your own level of understanding of the material. You are not allowed to talk with your neighbor or look at his exam sheet. Failure to obey this rule will simply result in a **zero score**.
- If you finish early, you can hand in the exam and leave early. However, this is only possible until at latest 15 minutes before the ending time of the exam. If less than 15 minutes are left, please stay seated and wait for the end.

STOP! Do not turn this page until the instructor tells you to do so.

Do NOT write in this section.

Problem	Max	Points
1	12	
2	14	
3	4	
4	10	
5	10	
Total	50	

Important: Verify that your exam book has 18 pages.

Problem 1: Multiple-choice questions (12 points)

Please answer the following questions by ticking the choices that apply. Each question has 1 point, and is clearly marked as a C, C++, or Python question. Note that multiple choices are possible for each question. Mark all possible choices that apply by ticking the corresponding box ☑ .

Question 1 (C): Assuming a 3-dimensional array is declared as **int ar[5][4][3]**, what is the correct syntax to send the 3-dimensional array as a parameter to a function **F( )**?

□ **F(ar);**
□ **F(&ar);**
□ **F(*ar);**
□ **F(**ar);**

Question 2 (C): What will be the output of the following C code, supposing that the corresponding executable file is run without any command line argument.

```
#include <stdio.h>
int main(int argc, char *argv[])
{
   while (argc--) {
      printf("%s\n", argv[argc]);
   }
   return 0;
}
```

□ A compile time error.
□ The name of the executable file.
□ A segmentation fault.
□ None of the above.

Question 3 (C): Consider the following declaration statement in C:

```
int *ptr, p;
```

Choose all of the following statements that are **not** correct!

□ **ptr** is a pointer to integer, but **p** is not.
□ Both **ptr** and **p** are pointers to integer.
□ **ptr** is a pointer to integer, but **p** may or may not be.
□ Neither **ptr** nor **p** is a pointer to integer.

Question 4 (C): What will be the output of the following C code?

```
#include <stdio.h>
int main(void)
{
    int a = 5;
    int *p = &a;
    int **q = &p;
    (**q)++;
    printf("%d\n", a);
    return 0;
}
```

□ 5
□ The address of **p**.
□ 6
□ Compile time error.

Question 5 (C++): Polymorphism is a programming concept that may …

□ … permit the same name to be used in different ways.
□ … be realized by overloading functions.
□ … be realized by virtual functions that are overridden in a child class.
□ … be realized through the application of templates.

Question 6 (C++): The default constructor …

□ … will not be provided by the compiler if no other constructor is defined.
□ … takes no arguments.
□ … is necessary if the class is to be used as a template parameter to a **std::vector** class.
□ … can't perform custom initializations of member variables inside its implementation body.

Question 7 (C++): The data inside a **std::map** is …

□ … stored by key-value pairs.
□ … stored using **std::pair**.
□ … having constant access time.
□ … having O(log(n)) access time.

Question 8 (C++): Concurrency: Mark all statements that apply

□ Memory can be shared between different processes.
□ Memory can be shared between different threads.
□ Non-atomic access to memory shared between different threads must be concurrent.
□ Access to memory shared between different threads can be made mutually exclusive by applying **std::mutex**.

Question 9 (Python): What will be the output of the following Python code?

```
x = 2
def foo():
    x = 3
    def bar():
        x = 4
        return lambda z: x+z
    return lambda z:bar()(x*z)
print(foo()(x))
```

□ 10
□ 11
□ 14
□ 24

Question 10 (Python): What will be the Method Resolution Order (MRO) of the class C4 in the following Python code?

```
class C1:
    def foo(self):
        print(1)
class C2(C1):
    None
class C3():
    def foo(self):
        print(3)

class C4(C2,C3):
None
```

□ C4 C3 C2 C1
□ C4 C2 C3 C1
□ C4 C2 C1 C3
□ C4 C3 C1 C2

Question 11 (Python): What will be the output of the following Python code?

```
def foo(a,b):
    s = 0
    try:
        s = s + bar(a,b)
        s = s + 10
    except ZeroDivisionError:
        s = s + 20
    else: s = s + 30
    finally: s = s + 40
    return s

def bar(a,b):
    s = 0
    try:
        a//b
        s = s + 1
    except ZeroDivisionError:
        s = s + 2
    else: s = s + 3
    finally: s = s + 4
    return s

print(foo(1,0)+foo(1,1))
```

□ 154

□ 156

□ 158

□ 174

Question 12 (Python): Choose all the function definitions that are correct in Python?

□
```
def foo(x, y, z):
    None
```

□
```
def foo(x:int, y, z:float):
    None
```

□
```
def foo(x, y=0, *z):
    None
```

□
```
def foo(x, **y, *z):
    None
```

Problem 2: Find and correct problems and bugs (14 points)

Question 1 (C, 4 points): Find 4 lines with errors in the following C code. State the line numbers to indicate errors. In addition, for each line number, briefly explain what the error is, and propose a correct replacement statement. Note: There could be more than 4 lines with errors, but you only need to find and correct 4 of them.

```
/* The following C program processes an array of student records. For
each record, it stores the student’s id and name. In the program, it reads
each student’s information, and then prints the student information. The
functions input() and output() are used for the reading and printing of
student information. */

1  #include <stdio.h>
2  #define SIZE 10
3  struct Student {
4     int id;
5     char name[10];
6  };
7  void input(struct Student s);
8  void output(struct Student s);
9  int main() {
10    struct Student s[SIZE];
11    int i;
12    for (i=0; i < SIZE; i++)
13    input(s[i]);
14    for (i=0; i < SIZE; i++)
15       output(s[i]);
16    printf(“\n”);
17    return 0;
18 }
19 void input(struct Student s) {
20    printf(“Student ID: ”);
21    scanf(“%d”, &s.id);
22    printf(“Student Name: ”);
23    scanf(“%s”, s.name);
24 }
25 void output(struct Student s) {
26    printf(“%s(%d) ”, s.name, s.id);
27 }
```

Question 2 (C++, 4 points): Find two problems in the following C++ code. Mention the relevant line, provide an explanation to the problem, and propose a solution.

```
1  class IntegerArray {
2  public:
3     int *m_data;
4     int m_size;
5
6     IntegerArray(int size) {
7        m_data = new int[size];
8        m_size = size;
9     }
10
11    ~IntegerArray() {}
12 };
13
14 int main() {
15    IntegerArray a(2);
16    a.m_data[0] = 4; a.m_data[1] = 2;
17    if (true) {
18       IntegerArray b = a;
19       b.m_data[0] = 6;
20    }
21    printf("%d\n", a.m_data[0]);
22 }
```

Question 3 (Python, 6 points): Find 6 errors in the following Python code. For each error, indicate the line number, briefly explain what the error is, and correct the statements. Note: There might be more than 6 errors, but you only need to find and correct 6 of them. The program is a grade evaluation program. It first reads and stores each student's name and score of an exam, then computes the average score of all scores, and finally stores the average score in each student's information.

```
1  def GetStudentsInfor(L = []):
2      '''Get all students' names and scores'''
3      for i in range(1,100):
4          L.append(GetOneStudentInfor())


5  def GetOneStudentInfor():
6      '''Get one student name and score'''
7        name = input("Please input name:")
8      score = input("Please input score:")
9      return (name,score)

10 def ComputeAverageScore(L):
11     '''Compute the arithmetic average of all students' scores'''
12     totalScore = sum( s[-1] for s in L)
13     averageScore = totalScore/100
14     return averageScore

15 def AddAverageScore(L,averageScore):
16     '''Add the average Score into each student's information'''
17     for i in L:
18         i.append(averageScore)

19 def GetInfor(L,studentName):
20     '''Get information of a student whose name is studentName'''
21     for i in L:
22         if i[0] is studentName:
23             return i
24 L = List()
25 GetStudentsInfor(L)
26 aver = ComputeAverageScore(L)
27 AddAverageScore(L,aver)
28 print(GetInfor(L,"foo"))
```

Problem 3: C problem (C, 4 points)

Write a **recursive** C function `countZeros()` that counts the number of zeros in a specified positive integer `num`. Assume that the left-most (also known as the most significant) digit is non-zero. The function passes the result to the caller through the parameter `count`, which is a pointer to an integer that is initialized to 0. For example, `countZeros(150060, count)` returns 3 and `countZeros(15867, count)` returns 0. The function prototype is given as follows:

```
      void countZeros(int num, int *count);
```

Note: You do not need to write the function `main()` which would test the function `countZeros()`.

Problem 4: C++ problem set (C++, 10 points)

Question 1: Predict the output of the following code. What code changes are needed if we additionally want to print full-time employees first. Note: You do not copy the entire code, you only need to write the new ordering function! (C++, 4 points)

```
#include <iostream>
#include <algorithm>

struct Employee {
public:
  std::string name, id;
  bool isFulltime;
};

bool comparator( Employee * op1, Employee * op2 ) {
  if( op1->id.size() != op2->id.size() ) {
    if( op1->id.size() < op2->id.size() )
      return true;
    return false;
  }
  return op1->id < op2->id;
}

int main() {
  Employee e1, e2, e3, e4;
  e1.name = "Cindy";           e1.id = "20180915";       e1.isFulltime = false;
  e2.name = "Simon";           e2.id = "20180309";       e2.isFulltime = false;
  e3.name = "Jennifer";        e3.id = "370";            e3.isFulltime = true;
  e4.name = "Paul";            e4.id = "20181002";       e4.isFulltime = true;

  std::map<std::string,Employee*> staff;
  staff["Cindy"]      = &e1;       staff["Simon"]        = &e2;
  staff["Jennifer"]   = &e3;       staff["Paul"]         = &e4;

  std::vector<Employee*> reorderedStaff( staff.size() ); int i = 0;
  for(   std::map<std::string,Employee*>::iterator it = staff.begin();
         it != staff.end(); it++ )
    reorderedStaff[i++] = it->second;
  std::sort( reorderedStaff.begin(), reorderedStaff.end(), comparator );
  for(   std::vector<Employee*>::iterator it = reorderedStaff.begin();
         it != reorderedStaff.end(); it++ )
    std::cout << (*it)->id << ": " << (*it)->name << std::endl;
}
```

Question 2: (C++, 6 points) Propose a template class that generalizes the below two classes. Add a function that allows us to write the sum of two vectors using the expression

```
DoubleVector d3 = d1 + d2;
```

and not

```
DoubleVector d3 = d1.sum(d2);
```

```
class DoubleVector {
public:
  DoubleVector( int size ) {
    m_size = size;
    m_data = new double[size];
  }

  virtual ~DoubleVector() {
    delete[] m_data;
  }

  double & operator[]  ( int index ) {
    return m_data[index];
  }

  DoubleVector sum( DoubleVector & op ) {
    DoubleVector result( this->m_size );
    for( int i = 0; i < this->m_size; i++ )
      result[i] = (*this)[i] + op[i];

    return result;
  }

private:
  double * m_data;
  int m_size;
};
```

```
class IntVector {
public:
  IntVector( int size ) {
    m_size = size;
    m_data = new int[size];
  }

  virtual ~IntVector() {
    delete[] m_data;
  }

  int & operator[]  ( int index ) {
    return m_data[index];
  }

  IntVector sum( IntVector & op ) {
    IntVector result( this->m_size );
    for( int i = 0; i < this->m_size; i++ )
      result[i] = (*this)[i] + op[i];

    return result;
  }

private:
  int * m_data;
  int m_size;
};
```

Problem 5: Python problem (Python, 10 points)

Read the following Rational Class and gcd function.

```
def gcd(m, n):
      if n == 0: m, n = n, m
      while m != 0: m, n = n%m, m
      return n

class Rational:
   def __init__(self, num, den = 1) :
      if den == 0 : raise ValueError
      sign = 1
      if (num < 0) :
         num, sign = -num, -sign
      if (den < 0) :
         den, sign = -den, -sign
      g = gcd(num, den)
      self.num = sign * (num//g)
      self.den = den//g
   def __str__(self) :
      return str(self.num) + "/" + str(self.den)

r = Rational(2,4)
print(r)  # outputs 1/2
```

1) Write a function for the class Rational such that the following code outputs 18/65 (4 points)

```
x = Rational(2,26)
y = Rational(5,25)
print(x+y)
```

2) Write a function for the class Rational such that the following code outputs 1/5 (6 points)

```
x = Rational(2,26)
y = Rational(5,25)
print(x*y*13)
```

(end of exam)