The University of Calgary
Faculty of Science
DEPARTMENT OF COMPUTER SCIENCE
Winter 2008
CPSC 219 Midterm Exam
Lecture 01

Student Name: _____

Student ID: _____

Instructions:

Please fill in your name and ID number above. Answer all questions in the space provided in this exam. At the end of the exam, turn in this exam booklet.

This is a closed book exam; no notes allowed. Calculators and electronic devices are not allowed. Please ensure that your cellphone is turned off.

Duration: 120 minutes.

Grading:

Each question will receive a letter grade. The final grade will be computed by summing the GPA of each question multiplied by its weight.

	Weight
Q1:	15
Q2:	5
Q3:	10
Q4:	5
Q5:	5
Q6:	10
Q7:	5
Q8:	10
Q9:	10
Q10:	10
Q11:	15

1. Identify 5 differences in syntax between python and C. Describe each with at least 1 sentence and provide an example which shows both the C and python versions. (15%)

2. Why is it not necessary to indent your code in C? (5%)

3. What is a potential hazard of the following C code? (10%)

```
char *strcpyX(char *s1, char *s2)
{
  char *temp = s1;
  while (*s2)
    *s1++ = *s2++;
  *s1 = '\0';
  return temp;
}
```

4. Identify three differences between lists in python and arrays in C. (5%)

5. Why is dynamic memory allocation important? (5%)

6. What is the difference between a “struct” definition in C and an abstract data type? (10%)

7. When we talk about structures, abstract data types and objects, we often use the word “instance”. What is an instance? (5%)

8. Why does the following code (described in class) produce random output? (10%)

```
struct address
{
  char street1[50];
  char street2[50];
  char city[50];
  char prov[50];
  char postal_code[50];
};

struct address *new_address()
{
  struct address temp;
  temp.street1[0] = '\0';
  temp.street2[0] = '\0';
  temp.city[0] = '\0';
  temp.prov[0] = '\0';
  temp.postal_code[0] = '\0';

  return &temp;
}

main(int argc, char **argv)
{
  if (argc<6)
    {
      printf("Usage: %s street1 street2 city prov postal_code\n", argv[0]);
      exit(1);
    }

  struct address *an_address = new_address();
  strcpy(an_address->street1, argv[1]);
  strcpy(an_address->street2, argv[2]);
  strcpy(an_address->city, argv[3]);
  strcpy(an_address->prov, argv[4]);
  strcpy(an_address->postal_code, argv[5]);

  printf("Address:\n\t%s\n\t%s\n\t%s\n\t%s\n\t%s\n", an_address->street1,
         an_address->street2, an_address->city, an_address->prov,
         an_address->postal_code);
}
```

9. How is object oriented analysis and design different from procedural decomposition? (10%)

10. Describe the process that one should follow to produce a class model from a problem. (10%)

11. Given the following linked list: (15%)

```
struct frequency
{
  char name[50];
  int count;
  struct frequency *next;
};

struct frequency *head;
```

write a function which takes a single string parameter. The function should traverse the linked list finding the node whose "name" matches the string parameter. When that node is found, increment the node's "count" variable by one.