CSCI 135 - Test 2

Name (first, then last):

EMPLID:

Morning Section: Evening Section:

Instructions

- READ THESE DIRECTIONS!
- Clear your desk before the test (including any cellphones, electronic devices, etc.)
- Do not open this exam before you are told to do so.
- This is a closed-book closed-notes test.
- You may not use constructs or library functions not covered in class or specific to some C++ version.
- For questions needing I/O, you do not need to output prompts and may assume all input is legal (except as otherwise stated)
- You will not be graded on minor syntax errors as long as they don't make your answer ambiguous. Conversely, conceptual errors result in major deductions, even if syntactically minor.
- Budget your time well. The questions are not necessarily in order of difficulty or time!

1. (12%) Short answer. Please keep answers short (1 sentence), as extra incorrect information counts negatively.

   (a) Define in your own words *information hiding*.

   (b) What are the object oriented analogs of the non-OO concepts of types and variables, respectively?

   (c) What does *garbage* mean (in the context of programming)?

   (d) Why can't you return an array from a function in C/C++?

2. (6%) Consider the following code fragment:

```
int data[SIZE];
int *datap = data;
data[k] = data[k-1];  // assume no bounds errors
```

   Rewrite the last line to use datap instead of data. You may not use any array operators such as indexing.

3. (20%) Consider the following fragment:

```
class Data {
public:
  Data(double newd, int newk);
  ~Data();
  double getd();
  int getk();
  void setd(double newd);
  void setk(int newk);
private:
  static int num = 0;
  double     d;
  int        *k;
};
Data s1;
Data *s2;
// fragment for last part goes here
```

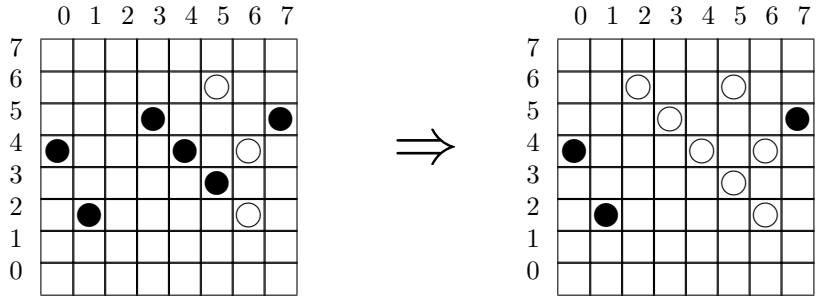
(a) Assuming ints, doubles, and pointers require 2, 4, and 6 bytes respectively, How much memory is allocated after executing the above code? Show your work!

(b) Write the code for the constructor whose prototype is given above (the arguments correspond to the members). It should not change the `num` member.

(c) Write a destructor appropriate for the class.

(d) Write a code fragment that 1) initializes s1 so that its d and k members are 3.14 and 7 respectively, and 2) the same to s2.

4. (21%) Consider a game on a SIZE × SIZE board, where each cell may be either empty or occupied by a black or white piece. In this game, when a cell is placed on the board, every cell of the opposite color in the southeast direction up to the first cell that is either empty or the same color changes color. For example, with SIZE = 8, the board is transformed as illustrated below if a white piece is placed at row 6 column 2; no existing piece changes color if a black piece is placed at row 6 column 6 or row 7 column 6.



Write a function `move` that places a cell of color x on the board at position p, based on the declarations and prototype below.

```
enum Cell {EMPTY,BLACK,WHITE};
Cell board[SIZE][SIZE];  // left dimension is row
struct Position {
  int row;
  int col;
};
Position pos;
void move(Cell x, Position p);
```

5. (20%) This problem **must** be done using recursion (no credit otherwise). The only string library functions you may use are: indexing, `length()`, `+`, and `substr(int n)` (recall that substr(n) returns a string with all but the first n characters).

   (a) Initially (at hour 0), there are `48` bacteria in a test tube, and the number doubles each hour. Additionally, `numnew` bacteria are added each hour. For example, with `numnew=1`, there are 97 bacteria at hour 1. Write a function that returns the number of bacteria at hour `h`, using the prototype: `int numbac(int h, int numnew)`

   (b) Write a function that returns a string with each instance of an argument character removed, using the prototype `string rem(string s, char c)`. For example, it would return ``bc'' if `s=``abac''` and `c='a'`.

6. (21%) Consider the following definitions:

```
class Student {
  string getFirst();
  string getLast();
  int    getGrade();
  void   setFirst(string s);
  void   setLast(string s);
  void   setGrade(int g);
private:
  string firstName;
  string lastName;
  int    grade;
};
```

```
class Section {
public:
  int     numStudents();
  Student getNthStudent(int n);
  string  getCourse();
  string  getInstructor();
private:
  Student students[SIZE];
  string  course;
  string  instructor;
};
```

Write a function, `honorRoll` that returns a dynamic array containing only those students in section `sec` with grades higher than `grade` (and sets the `size` of the returned array). Use the prototype:
`Student * honorRoll(Section sec, int grade, int & size)`
Hint: you will neeed to first determine how many students are on the honor roll.