CISC 181 Test 2 **April 24, 2003**

Name:_____

1. In the True/False questions if the answer is False, explain why.

   (a) (2 pts) The elements of an array are related by the fact that they have the same _____.

   (b) (3 pts) An m-by-n array contains _____ rows, _____ columns, and _____ elements.

   (c) (12 pts) For the following, write a single statement that performs the indicated task. Assume the following declarations:

   ```
   float number1 = 7.3, number2;
   char *ptr;
   char s1[100], s2[100];
   ```

      i. Declare the variable `fPtr` to be a pointer to an object of type `float`.

      ii. Assign the address of variable `number1` to pointer variable `fPtr`.

      iii. Print the value of the object pointed to by `fPtr`.

      iv. Assign to the object pointed to by `fPtr` the value of the variable `number2`.

      v. Print the value of `number1`.

      vi. Assume strings `s1` and `s2` have been assigned values. Print `EQUALS` if the string `s1` is the same as the string `s2`.

   (d) (2 pts) Structure members are accessed via the _____ operator in conjunction with an object of the struct type.

   (e) (3 pts) Unlike arrays, entire structures of the same type may be compared using the logical operator `==`. **True False**

   (f) (3 pts) It is possible to have arrays of any type including arrays of pointers and arrays of structures. **True False**

   (g) (3 pts) Like a structure, the declaration of a class creates a new data type but does not create variables of that type. This is done in a separate declaration similar to what is done with a structure. **True False**

   (h) (2 pts) Members of a class specified as _____ are accessible only to member functions of the class and friends of the class.

   (i) (2 pts) A _____ function is a special member function used to initialize the data members of a class.

   (j) (3 pts) Correct any errors (assuming proper include statements) in the following statements:

   ```
   int k; cin >> k;
   double* B = new int[k];
   ```

(k) (3 pts) Suppose `salary.data` is a file of floats. Correct any errors in the following statements (assuming proper include statements and assuming that the file is opened without any problems).

```
ifstream fin("salary.data");
float nextSalary = 0;
cin >> nextSalary;
```

(l) (4 pts) Consider the function prototype with a call-by-reference parameter
`void bar(int& a);`
and the function call
`int b; bar(b);`
Rewrite both lines of code, using (i) a single parameter that is called by value in the function prototype and (ii) pointers, to get an effect equivalent to the code above.

2. (18 pts) Consider the following program.

```
#include<iostream.h>

double* test(double A[], int n);

main()
{
 double Number[] = {0.2, 2.2, 13.1, -1.5, 4.6, 7.3};

 double *ptr = test(Number,6);

 cout << "Test = " << *ptr << endl;
}

double* test(double A[], int n)
{
  double *mAddr = &A[0],
         *endAddr = &A[n];

  for (double *iAddr = &A[1]; iAddr < endAddr; iAddr++)
    if (*iAddr > *mAddr) mAddr = iAddr;

  return mAddr;
}
```

(a) The type of the variable `Number[2]` is _____.

(b) The type of the variable `ptr` is _____.

(c) The type of the variable `endAddr` is _____.

(d) The type of the constant `&A[1]` is _____.

(e) What output is produced by this program?

(f) Give an overall, gestalt description of the task performed by the function `test()`.

3. (15 pts) Write a program that reads strings (until an EOF) and prints each string that begins with the three characters `You`. Assume that each string contains at most 50 characters.

4. (15 pts) Below is a declaration for the structure `Automobile`. Using it, write a function

```
int countAutos( Automobile A[], int n, Style bs, Year y )
```

that returns a count of the number of automobiles in the array A of $n \geq 0$ automobiles that have body style `bs` and year `y`. For example, a call like the following
`countAutos( A, 100, CONVERTIBLE, 2003 )`
would return the number of 2003 convertibles among the 100 cars A[0], A[1], ..., A[99].

```
enum Make {CHEVROLET, FORD, HONDA, ROLLSROYCE, YUGO};

enum Color {BLUE, RED, SILVER, TEAL, WHITE};

enum Style {CONVERTIBLE, SEDAN, SPORTS, WAGON};

struct Automobile
{
  Make make;
  int year;
  Style body;
  Color clr;
  int doors;
  int licenseNo;
  int cost;
};
```

5. (10 pts) Consider the following program

```
#include <iostream.h>
#include <string.h>

class BankAcct{
   public:

   BankAcct(char firstN[] = "", char lName[] = "", double bal = 0.0);
   char* getfirstName();

   private:

   char firstName[21];
   char lastName[21];
   double balance;
};

   BankAcct::BankAcct(char firstN[], char lastN[], double bal){
      strcpy(firstName, firstN);
      strcpy(lastName, lastN);
```

```
   balance = bal;
}

char* BankAcct::getfirstName(){
   return firstName;
}
```

Write a `main()` program that creates

(a) a variable `one` of type `BankAcct` with default initialization;

(b) a variable of type `BankAcct` with name Monica Henzinger and a balance of $500;

(c) a variable `three` of type `BankAcct` with name Barry Trager and a balance of $0.00; and then

(d) prints the first name for each account created.

Uses the fewest args possible in declaring the variables.