

EEL 3801C – Final examination

Date: April 20th, 2004

Name:

Student Code:

Instructions:

- This exam is open book and open notes. Allotted time is 90 minutes.
- Add comments to your code!!! Uncommented and incorrect code will not get partial credit.

Problem 1 (20 points)

Let ARRAY be an array of 8 bit integers, and LENGTH the number of integers in the array. One example is:

```
ARRAY    db      4, 2, 5, 6
LENGTH  dw          4
```

Write an assembly language program, which increments with 1 every number in the array and writes it back to the array. For example, after the program runs, the array above will contain the numbers 5, 3, 6, 7.

Assume that no overflow will happen and all the results fit in 8 bits.

Problem 2 (20 points)

Implement problem 1 in the C language.

Problem 3 (20 points)

Consider the following structure:

```
struct Animal{  
    char *name;  
    int legs;  
}
```

In your main function:

- a) Create a local variable of type `Animal`, and initialize it such that it contains the description of a horse.
- b) Create a pointer to the type `Animal`. Allocate memory for the variable, and initialize it to contain the description of a chicken.
- c) Before the termination of the main function, deallocate all the dynamically allocated memory.

Problem 4 (20 points)

a) Create classes to represent the following entities:

- Person: has a name.
- Student: is a type of a person who also has a school.
- Senior: is a type of student who also has a senior design topic.

Hint: you will need to use inheritance for this.

b) In a main function, create and initialize one example of each of the classes.

Problem 5 (20 points)

- a) We want to write a function `Half` which halves the value of the parameter passed to it. For example, if the value of variable `X` is 6, after calling `Half(X)`, the value of `X` will be 3. Because the variable `X` can be of different types, implement the problem through overloaded functions for the types `int`, `float` and `double`.
- b) Solve part a) of the problem through the use of C++ templates.

Bonus Problem 6 (0..50 points)

Do this problem last. This is an open-ended problem, similar to those you will encounter in your professional career. You don't need to solve it completely. You will get progressively more points depending on how complete your solution is.

Design a class called RestaurantBill. Here is a starting point:

```
class RestaurantBill {
public:
    int value;
    int tip;
    void printBill();
}

void RestaurantBill::printBill() {
    cout << value << tip;
}
```

This problem asks you to extend the class such that it becomes actually useable for the accounting of a restaurant. For example:

- the implementation does not handle cents (you need to count in dollars)
- the printing function does not print in a format similar to a regular bill
- it does not allow for the list of individual items, their names and prices
- it doesn't handle the tax
- it would be useful if it would suggest an amount for the tip
- it would need some sensible constructors
- it should include the name of the waiter

- a) Extend the class to implement as many of these functions as possible.
- b) Suggest new useful functionalities for this class.