

Name: key

PLEASE PRINT CLEARLY

ECE x70 Exam No. 1 (100pts. - 25% of the final grade)

General Remarks

This is in-class one-hour-long exam. You can use your notes, textbook, and any existing Web-based resources. You can use a lab or your own computer but must not use a cell phone. You must not communicate with other people or post the questions on an Internet forum. Provide a concise answer and to the point for maximum credit. Answers that are too long take too much time and may indicate that the author is unable to rank the importance of facts.

DL: ERR: PTS:

DL – exam difficulty level (adjustment), ERR – exam errors, PTS – exam point

Problem 1A (20pts.)

Consider the following function FN, and a main program that utilizes it.

Assume no pointer or other typecasting errors, and that operating system does not catch out of range memory access. Indicate what will be printed. Use “???” for unpredictable values.

```
#include <iostream>
static int d = 0;

void FN(int a, int &b, int* c) {
    d = *c;
    a++; b++; (*c)++; c++;
    static int d = a;
    d++;
    b = d;
}

int main() {
    int a = 12, b = 13, c = 14, d = 15;
    FN(d, c, &b);
    std::cout << a << " " << b << " " << c << " " << d << " ";
    FN(c, b, &a);
    std::cout << a << " " << b << " " << c << " " << d << std::endl;
    return(0);
} /* PRINTED:
```

12 14 17 15 13 18 17 15
 Note: some fields may remain blank, each mistake costs 2.5 points until 20 */

Problem 1B Quick Questions (5pts.)

Answer the questions either by writing in an answer into a blank, or circling correct one from the list:

- The range of values that can be stored in char variable is: between and or undefined
- The value of expression `0.0*sqrt(-1.0F)` computed using the IEEE standard float is

Answer: 0 Inf NaN/Ind undefined-behavior

- The loop `for(short int x=1000; x>0; x++);`

Answer: runs_forever ends undefined

Total errors this page:

Problem 2 (17pts.)

Analyze and complete a partially written program that asks user for a name of a file to read, opens the file, and repetitively reads line by line from the file, removes any initial white space (' ' or '\t'), and prints it to cout. File name may include space. Some blanks may remain empty.

```
#include < fstream >
#include <iostream>
#include < string >
using namespace std;
int main() {
    string fname;

    cout << "Name of the file to read from: ";

    getline( cin, fname ) ;

    ifstream file;

    file. open ( fname );

    if ( file.fail() ) { cerr << "cannot open" << endl; return(1); }

    while(true) {

        string buffer;

        getline ( file , buffer ) ;

        if ( file.fail() ) break;

        size_t pos = buffer. find_first_of ( " \t", 0 );

        buffer. erase ( 0 , pos );

        cout << buffer << endl;

    }
    file.clear() ;

    file.close() ;

    return(0);
}
```

Problem 3A (18pts.)

Implement the function that extracts the user ID from the first occurrence of an email address. An email address starts with the character sequence “mailto:” followed by the user ID (to be extracted), and then by character ‘@’ and the remainder of the email address. A user ID would not contain ‘@’. In case “mailto:” cannot be found or is not followed by ‘@’ in the remainder of the string then the function shall return an empty string. You must utilize the partially written code in your answer.

```
string get_user_id(const string & data) {
    size_t pos1 = data.find("mailto:", 0);
    size_t pos2 = string::npos;

    if ( pos1 != string::npos )

        pos2 = data.find( '@' , pos1 + 7 );

    if ( pos1!=string::npos && pos2!=string::npos )

        return( data.substr( pos1 + 7 , pos2 - pos1 - 7 ) );

    else

        return( string( "" ) );
}
```

Problem 2B+3B Quick Questions (10pts.)

Consider string variable *string STR*;

The function call <i>STR.substr(...);</i>	changes	<u>does not change</u>	contents of STR.
The function call <i>STR.erase(...);</i>	<u>changes</u>	does not change	contents of STR.
The function call <i>STR.find(...);</i>	changes	<u>does not change</u>	contents of STR.
The function call <i>STR.empty();</i>	changes	<u>does not change</u>	contents of STR.

Consider need for calling function *ifstream::close()*, which one below is correct:

- ☐ the explicit function call is never necessary because file object destructor closes the file anyway
- ☒ the explicit function call is really needed only if the file is to be reopened again later on
- ☐ the explicit function call is always needed because, otherwise, the file may remain locked as used

Problem 4A (20pts.)

The following problems apply to the partially defined class examples that were used in a recent homework. Write implementations for the indicated class methods that implemented functionality described in the comments to the code in a way similar to demonstrated in the Timer class example.

```

/* THE HEADER FILE - CITY.h *
class CITY {
private:
    double urb, sub, exu;
public:
    CITY(const double urban, const double suburban, const double exurban) const
        : urb(urban), sub(suburban), exu(exurban) {}

    friend bool LargerThan ( const CITY &one, const CITY &two) const ;
    // TO DO 1: returns true if the city "one" has larger
    //           total population than the city "two", false otherwise - in .CPP

    double getTotalPopulation( const ) const ;
    // TO DO 2: returns the sum of populations for all zones - inline in .H

    void clear( const ) const ;
    // TO DO 3: set all internal zone population variables to zero - decide where

    // TO DO 4: cross out the word const from the places where it does not belong
};
// TO DO 2: is implemented below:

inline double CITY::getTotalPopulaiton() const { return( urb+sub+exu ); }

inline void CTY::clear() { urb=0.0; sub=0.0; exu=0.0; } // the best place

```

Your implementation file:

```

/* THE LIBRARY IMPLEMENTATION FILE - CITY.cpp *
#include "CITY.h"

bool LargerThan( const CITY &one, const CITY &two ) {
    return( one.getTotalPopulaiton() > two.getTotalPopulaiton() );
    // return( one.urb+one.sub+one.exu > two.urb+two.sub+two.exu );
}

```

End of test

Total errors this page: ____