CSE278 SYSTEMS 1

Department of Computer Science and Software Engineering
Miami University

Final Exam (Section C & D)

Spring 2020

May 13, 2020
Full Marks: 100
There are 13 Questions for a total of 100 points.
Answer ALL the questions

NAME: William Mechler **MUID: mechlewg**

General Instructions

- First, ensure you have all the 15 pages of this exam booklet even before starting
- This exam is closed notes and closed books. No discussions are permitted
- You may use Calculator
- Do not bring out your cell phone; don't answer the phone; don't read text messages
- You have 2 hours to complete the exam
- Write your answers clearly
- The size of the space given for each answer is sufficient
- Write no more than 3-4 lines for each of the short questions
- Show all your works for the Mathematical problems
- Even if your final answers are incorrect, you will get partial credit if intermediate steps are clearly shown to highlight thought process. This applies to program tracing questions as well.

DISCLAIMER: By submitting the completed pdf file, you are pledged that you did not consult any one during the Exam time and everything was closed books and closed notes.

GOOD LUCK!

1. **(10 pt) Systems Security**

   a. List and briefly define categories of security mechanisms.

- Computer security
  Deals with programs and users on a computer and also deals with viruses, worms etc.

- Network security
  Deals with various layers in a network including protocols.

- Cryptography
  Deals with encrypting / decrypting information to avoid malicious users from accessing data.

- Physical security
  Securing data centers and physical facilities.

   b. The list of all passwords is kept within the operating system. Thus, if a user manages to read this list, password protection is no longer provided. Suggest a scheme that will avoid this problem. (Hint: Use different internal and external representations.)

One why is to have the user validate themselves before they are given access to this list.

   c. What is the difference between using a packet filter and using an application-level gateway approach in a firewall?

Packet filter firewall are less secure than application level firewalls because the packet filter doesn't understand application layer protocols.

   d. What is stored in the `.eh_frame` section in an ELF file?

The .eh_frame contains exception unwinding and source language information.

   e. What is the difference between *segments* and *sections* in ELF ?

Segments contain information that is necessary for runtime execution of the file while sections contain important data for linking and relocation.

2. [6 pt] (***Find Errors in code***)

Assume the following declarations and statements:

```
int *zPtr;           //  zPtr will reference built-in array z

void *sPtr = nullptr;

int number;

int z[6] = { 10, 20, 30, 40, 50, 60};.
```

a)

```
zPtr = z ;
```

b)

```
// use pointer to get second value of a built-in array
            number = *(zPtr + 2);
```

c)

```
// assign built-in array element 3 (the value 40) to number
            *(zPtr + 3) = number;
```

d)

```
// display entire built-in array z in reverse order
            for( int  i = 5; i >= 0; i--)
                        cout << *(zPtr + i) << endl;
```

e)

```
// assign the value pointed to by sPtr to first element of z
          z[0] = *sPtr;
```

f)

```
z++;
```

3. (**10 pt**) Computer Architecture

A. Elaborate the meaning of the following acronyms:

- ABI – Application Binary Interface
- ELF – Executable & Linkable Format
- GDB – Generic Data Base
- ALU - Arithmetic & Logic Unit.
- SMTP – Simple Mail Transfer Protocol
- GREP – Global Regular Expression and Print

B. (***Integer Equivalent of a Character***)

Write a complete C++ program that prints the integer equivalent of a character typed at the keyboard. Store the input in a variable of type `char`. (Make sure to use appropriate *type casting*)

```
#include <iostream>

using namespace std;

int main(){
char input;
cin >> input;
cout << static_cast<int>(input);

}
```

4. [5] C++ Fundamentals

A. In what way object-oriented analysis and design fundamentally different from other approaches?

B. Compare and contrast the `while` and `do . . . while` iteration statements.

A while will check before the first iteration starts and will check again before starting another iteration. When a do will only check after each iteration meaning it will also run at least once.

C. Why not make all classes friends of all the classes they use?

D. Consider the following command of compiling C++ code:

g++ -std=c++14 –g –Wall source.cpp –o nameOfExecutbale

What is the purpose of the flag g?

E. What are the two different ways to write comments in C++ code?

You can comment out one line of code by doing the following.

```
  // your code
```

Or you can make a multiline comment by doing the following.

```
/*
 Your code
  Your code
/*
```

5. [5] (***Factorials***)

Consider the following implementation of computing factorials *recursively*:

```
long recursiveFactorial (int number)
{
        if (number <= 0) {
                return 1;
        }
        else {
                return number * recurviseFactorial (number -1);
        }
}
```

Re-write the function using *iteration* only. What difficulty might prevent you from calculating the factorial of 100?

```
long Factorial (int number)
{
         int i, input;
        long fact;

        cin >> input;
        for( i = 1;  i <= input; i++){

           fact=fact*I;
         }
  return fact;
}
```

Because the factorial of 100 is a huge number it may exceed the limit of the amount of information that can be stored in a long.

6. [4] (***Pass-by-Value vs Pass-by-Reference with Pointer***)

Consider the following code snippet demonstrating Pass-by-Value.

```
#include <iostream>
using namespace std;
int cubeByValue(int);    // prototype
int main() {
        int number{5};
        cout << "The original value of number is  " << number;
        number =  cubeByValue (number);
        cout << "The new value of number is  " <<  number << endl;
}
int cubeByValue(int n)
{
return n * n * n;
}
```

Re-write the code to handle Pass-by-Reference with Pointer.

7. [10] (***Duplicate Elimination with vector***)

Use integer vector to solve the following problem. Read in 20 numbers, each of which is between 10 and 100 inclusive. Begin with an empty vector and use its `push_back` function to add each unique value to the vector. As each number is read, validate it and store it in the vector only if it isn't a duplicate of a number already read. After reading all the values, display only the unique values that the user entered.

8. [9] (**NumberSystems**)

Convert the following numbers from one base to another: (Show all of your mathematical works/steps)

- From 252 Decimal to Hexadecimal

252 decimal = FC Hexadecimal

- From F8C Hexadecimal to Decimal

F8C Hexadecimal = 3980 decimal

- From 88 Decimal to Binary

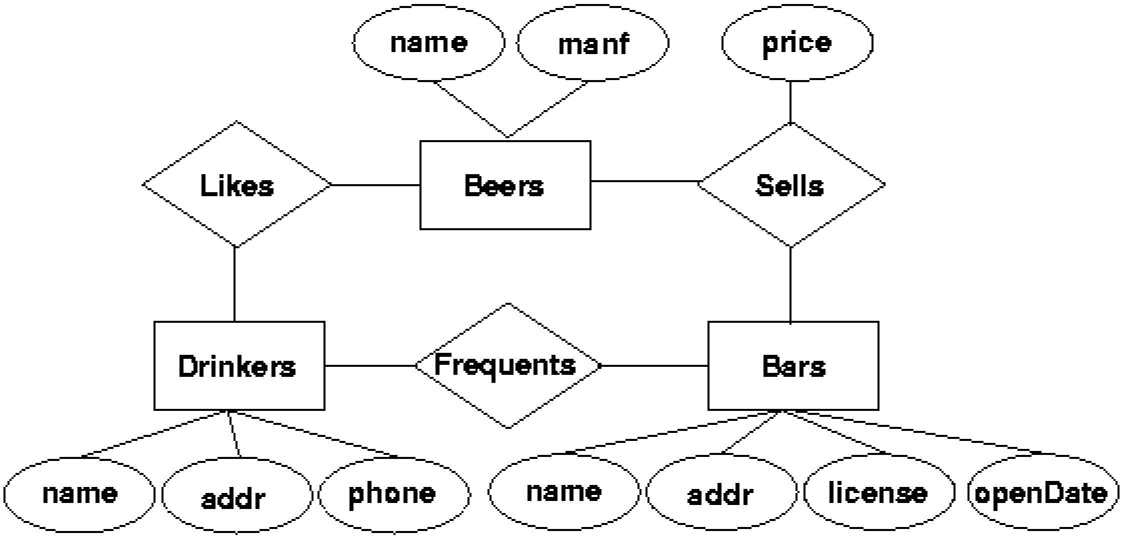
88 decimal = 1011000 binary

9. [15] DBMS

A. Define the *two principle integrity rules* for the relational model. Discuss why it is desirable to enforce these rules.

The two principle integrity rules are Entity integrity and Referential integrity. Frist for entity integrity it is important to make sure each entity is distinguishable and have unique identification of some kind. And referential integrity is important to make sure your relation between two entities does not break down.

B. A Toy database: ER schema



C.

A Toy database: relational schema

Drinkers (name, addr, phone)

Beers (name, manf)

Bars (name, addr, license, openDate)

Likes (drinker, beer)

Sells (bar, beer, price)

Frequents (drinker, bar)

The following notations apply: primary keys, foreign keys and primary keys which are also foreign keys

Corresponding Oracle like DDL

```
drop table Likes;
drop table Frequents;
drop table Sells;
drop table Beers;
drop table Bars;

drop table Drinkers;

create table Beers (
        Name varchar(30) primary key,
        Manf varchar(20)
);
create table Bars (
        Name     varchar(30) primary key,
        Addr     varchar(20),
        License  integer,
        openDate date
);
create table Drinkers (
        Name   varchar(20) primary key,
        Addr   varchar(30),
        Phone  char(10)

);

create table Likes (
        Drinker varchar(20),
        Beer    varchar(30),
        constraint LikesC1 primary key (Drinker,Beer),
        constraint LikesC2 foreign key (Beer)
                references Beers(name) on delete cascade,
        constraint LikesC3 foreign key (Drinker)
                references Drinkers(Name)  on delete cascade
);
create table Sells (
        Bar   varchar(30),
        Beer  varchar(30),
        Price real,
        constraint sellsC1 primary key (Bar,Beer),
        constraint sellsC2 check (Price >= 0),
        constraint sellsC3 foreign key (Beer)
                references Beers(Name),
        constraint sellsC3 foreign key (Bar)
                references Bars(Name)

);

create table Frequents (
        Drinker varchar(20),
        Bar     varchar(30),
        constraint FreqC1 primary key (Drinker,Bar),
        constraint FreqC2 foreign key (Bar)
                references Bars(Name),
        constraint FreqC3 foreign key (Drinker)
                references Drinkers(Name)
);
```

Answer the following SQL queries:

A. Retrieve *where* and for *which price* the beer ' Budweiser ' is sold

Select Bar, Price from Sells where Beer = ‘Budweiser’;

B. Retrieve the bars (name and address) which are located in ‘Oxford’

Select Name, Addr From Bars
    where Addr Like ‘%Oxford%’;

C. For each *bars* where ‘John’ is used to go, retrieve the *beers* which are served and for each its *price*.

Select Beer, Price From

D. For **Part C** of the above SQL query which is supposed to be executed and displayed in mysql> command prompt, **do the same this** time using C++ MySQL connectivity so that we can see the same result from Linux shell command prompt. For your conveniences, I am attaching some start up code as template that we used in Lab: Make necessary assumptions.

```
int main(int argc, char *argv[]) {
  mysqlpp::Connection myDB("cse278s19",
"os1.csi.miamioh.edu", "cse278s19", "rbHkqL64VpcJ2ezj");

  mysqlpp::Query query = myDB.query();
   //query << "SELECT pname, price FROM Product;";
   //query << "SELECT id, name, email from MUid where name
like '%%Doe%%';";
   query << "SELECT  * FROM Company;";
   query.parse();

   mysqlpp::StoreQueryResult result = query.store();

   for (size_t row = 0; ( row < result.size()); row++) {
        //int id = result[row][0];
        std::string cname = result[row][0].c_str();
        //std::string name = result[row][1].c_str();
        int stockPrice = result[row][1];
        std::string country = result[row][2].c_str();
        //float price = result[row][1];
       std::cout << cname << "\t" << stockPrice << "\t" <<
country << "\n";
   }
  return 0;
}
```

10. [5] (***Unordered_map***)

Complete the following method that reports the IP address given Domain name (order of entries is random). In case the domain name can't be resolved, return "Not Found" as a result. Make necessary assumptions. In the Lab we experimented using the Linux command dig with a number of domain such as following:

Domain	IP Address
www.miamioh.edu	10.5.32.12
www.gnu.org	209.51.188.148
www.gov.cn	65.153.158.199

```
using namespace std;

using DomainMap = std::unordered_map<std::string, std::string>;

std::string find(const DomainMap& m1, const std::string)
```

11. Using the following memory layout and symbol table, answer the following questions: [**5×2=10 points**]

Memory Layout:

Address

0x100	*0x101*	*0x102*	*0x103*	*0x104*	*0x105*
02	01	06	01	41	42

0x106	*0x107*	*0x108*	*0x109*	*0x10A*	*0x10B*
00	00	00	01		

Symbol Table (all `short`):

Symbol	*Address*
`var`	0x102
`empty`	0x106
`last`	0x104

a. Indicate the result of the following statements:

```
short *sp = &var;
short x   = *sp;
```

The value stored in variable `x` is:

b. Indicate the result of the following statements:

```
short *sp = &empty;
short x   = *sp;
```

The value stored in variable `x` is:

c. Indicate the result of the following statements:

```
short *sp = &var;
short x   = *(sp + 1);
```

The value stored in variable `x` is:

d. Indicate the result of the following statements:

```
char  *cp = reinterpret_cast<char*>(&var);
short *sp = reinterpret_cast<short*>(cp + 1);
short   x = *sp;
```

The value stored in variable `x` is:

e. Indicate the result of the following statements:

```
int getIt(short arr[]) {
        return arr[1];
    }

    short    x = getIt(&var);
```

The value stored in variable x is:

12. [6] (**Misc**)

A. What is the Linux command to record user activity?

The script command can be used to record user activity.

B. What is the semantic difference between the commands cp vs mv ?

The cp command is for copying files where the mv command is for moving files.

C. What is the difference between ptr, *ptr and **ptr ?

D. C++ supports all 4 number systems to represent numbers. How? Give some example.

E. What do you mean by Hardware Software Interface? Give a concrete example.

F. What was the name of the *linker file* to experiment with Networking and MySQL database connectivity by C++ code in os1.csi.miamioh.edu?

13. (5) What does this program do?

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;


void mystery1( char *, const char *);  // prototype


int main()
{
    char string1[ 80 ];
    char string2[ 80 ];

    cout << “Enter two strings: “;
    cin >> string1 >> string2;
    mystery1( string1, string2);
    cout << string1 << endl;
    return 0;  // indicates successful termination
}  // end main


void mystery1 ( char *s1, const char *s2 )
{
    while (*s1 != ‘\0’)
            ++s1;
    for ( ; *s1 = *s2;  s1++, s2++)

            ; // empty statement

}  // end function mystery1
```

ANS: The program has the user input two strings and then the mystery function combines the two strings.