

# CSci 1113

## Final

Name: \_\_\_\_\_

Student ID: \_\_\_\_\_

*Instructions:* Please pick and answer any 10 of the 12 problems for a total of 100 points. If you answer more than 10 problems, only the first 10 will be graded. The time limit is 120 minutes. Please write your answers in the space provided. The exam is open book and notes. You may use electronic devices to ONLY look at either an e-book version or electronic notes. You may not use the internet, compiler or any other outside resources. (If you are typing on your keyboard/input device for anything other than ctrl-F to find words in the e-book or notes, this is probably not acceptable.)

### **Problem (1)** [10 points]

Make a class (in C++) called “umnStudent” that stores three things: (1) a name, (2) an emailID and (3) your ID number. Then use a constructor to make a variable called “me” in main() that holds your student information (if you do not know your own student ID number, you can make it up).

**Problem (2)** [10 points]

Assume you have the “GPA” class as defined below. Write code (in C++) so you can add two GPAs together to get a new cumulative GPA. The formula for this is:  $\frac{average_1 \cdot credits_1 + average_2 \cdot credits_2}{credits_1 + credits_2}$ .

```
class GPA {  
private:  
    int credits;  
    double average;  
};
```

**Problem (3)** [10 points]

What is the output of the following code? Without inserting any lines in the middle of the given code, delete as many memory leaks as possible. How many memory leaks remain after your deletes? You must show work to receive full and partial and credit.

```
int ** dx;
int * sx;
int * sp;
int x = 4;

dx = new int*;
sx = new int;
sp = &x;
*dx = sp;
**dx = 7;
*dx = sx;
**dx = -8;
sx = new int;
*sx = 2;

cout << **dx << " " << *sx << " " << *sp << " " << x;
```

**Problem (4)** [10 points]

Given the class outline below which includes a destructor, write (in C++) a default constructor that ensures: (1) no memory leaks and (2) nothing is double deleted (or improperly deleted in any other way).

```
class ruhroh {
private:
    string **w;
    string *x;
    int* arr[100];
public:
    ~ruhroh();
};

ruhroh::~ruhroh() {
    for(int i=0; i < size; i++) {
        if(arr[i] != NULL) {
            delete arr[i];
        }
    }

    delete *w;
}
```

**Problem (5)** [10 points]

Write (C++) code for a function that takes two inputs (1) an array of “Date” objects (as defined below) and (2) the size of the input array. This function should return an array of integers, where the value at each index is the corresponding the year from the “Date” array.

```
class Date {
private:
    int month;
    int day;
    int year; // worst order ever btw
public:
    Date();
    Date(int m, int d, int y);
    int getYear();
};

int Date::getYear() {
    return year;
}
```

**Problem (6)** [10 points]  
What is the output of this code?

```
class A {
private:
    int x;
    int y;
public:
    virtual void foo();
    A();
};

class B : public A {
private:
    int x;
public:
    void foo();
    B();
};

class C : public A {
private:
    int x;
public:
    C();
};

A::A() {
    cout << "Heya!\n";
    x=1;
}
B::B() {
    x=2;
}
C::C() {
    cout << "Live long and prosper\n";
    x=3;
}

void A::foo() {
    cout << x << endl;
}
void B::foo() {
    cout << x << endl;
}
int main() {
    C d;
    B s;
    s.foo(); // B object
    d.foo(); // C object
}
```

**Problem (7)** [10 points]

Write (in C++) a copy-constructor for the class below. A constructor is provided for reference on how the variables are used, but these will not necessarily always be the values in the class.

```
class classic {
private:
    char* car;
    char* list;
public:
    classic();
};

classic::classic() {
    list = new char[10];
    for(int i=0; i < 10; i++) {
        list[i] = 'c';
    }
    car = &list[rand()%10];
}
```

**Problem (8)** [10 points]

Find 3 errors in the code below. Assume that the code is completely shown except for #includes and "using namespace std". For each error, identify whether it is a runtime error, syntax error or logic error. You must also precisely describe why you think the part of code you identify is an error.

```
class timer {
private:
    int tick;
public:
    void nextSecond() { // functions are in here to save space (but it works)
        tick++;
    }
    timer() {
        tick = 0;
    }
};
class stopwatch : public timer {
private:
    int stime;
public:
    void start() {
        stime = tick;
    }
    void stop() { // can assume only 1 stop per "start"
        return tick - stime;
    }
};
int main() {
    srand(time(0));
    stopwatch gogogo;
    int r = rand()%100;
    for(int i=0; i < r; i++) {
        gogogo.nextSecond();
    }
    gogogo.start();
    int r = rand()%100;
    for(int i=0; i < r; i++) {
        gogogo.nextSecond();
    }
    cout << "Timed at " << gogogo.stop() << " seconds!\n";
}
```

**Problem (9)** [10 points]

Open a file called “essay.txt” and count how many words there are in the file (you may assume spaces are the only thing that determines words). If you cannot open the file say this, otherwise tell how many words are in the file.

**Problem (10)** [10 points]

Write both a `main()` function and a `hasSeven()` function. In `main()`, make an array of 10 integers and properly initialize them to values between 0 to 9 (inclusive). Then use the `hasSeven()` function to determine if this array contains the number 7. You must pass the array (and size if you wish) into this `hasSeven()` function, which will return true if the number 7 is one of the random 10 numbers (and otherwise false).

**Problem (11)** [10 points]

Write code for the following function so that it removes all the letters 's' from the input string. A sample main is provided below to clarify the usage of this function.

```
int main() {
    string tt = "sally sells shells by the sea shore";
    stopTheInsanity(tt);
    cout << tt; // should show: ally ell hell by the ea hore
}
```

**Problem (12)** [10 points]

List or describe precisely all cin inputs that could potentially crash the following code:

```
int x, y;
cin >> x >> y;

int arr[1000];
for(int i=0; i < y; i++) {
    arr[i] = i*i;
}

if(x == 0 || y/x > 1) {
    cout << arr[x];
}
else {
    cout << arr[-y/x];
}
```