

CSci 1113

Final

Name: _____

Student ID: _____

Instructions: Please pick and answer any 10 of the 11 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]

Consider the outline to the “BankAccount” class below (data inside is self-explanatory). Add two functions to the BankAccount class, an “add()” function and a “withdraw()” function. Both of these functions should take a single number as an input (make sure it is positive). The “add()” function should be void and always increase the money by the amount input into the function. The “withdraw()” function should return a bool: true if there is enough money in the bank account and false if the money requested is more than is in the account (no loans). If the function returns false, the money in the account should not change.

```
class BankAccount {
private:
    string name;
    double money;
    Date DOB; //date of birth
    string email;
};
```

Problem (2) [10 points]

This problem continues off the “BankAccount” class from the previous problem. Write a non-default constructor (with 4 inputs) to initialize all the values. You may assume a “Date()” constructor exists that takes three ints as an input. In addition to making a constructor for “BankAccount”, write a short main() where you: Create an account for “JP” with 0 initial money, born on 1/2/1999, and email of “myspam@gmail.com”. Then add \$100 to JP’s account and attempt to withdraw \$150 (should return false). Finally print the remaining balance of the account (should be \$100 still, but you have to use the appropriate variable and not just: cout << 100).

Problem (3) [10 points]

What is the output of the following code?

```
class A {
public:
    int x;
    int y;
    void printA();
    A();
};

A::A() {
    x=1;
    y=2;
}

void A::printA() {
    cout << x << y;
}

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

B::B() {
    x = 9;
}

void B::printB() {
    printA();
    cout << x << y << endl;
}

int main() {
    B x;
    x.printB();
    A y;
    y.printA();
}
```

Problem (4) [10 points]

Modify the following code to still work, while not leaking memory. Feel free to write code and then draw an arrow to where it should go in the existing code. You must use the existing code and not try to rewrite something from scratch with the same functionality.

```
srand(time(0));
int* nums[100];
for(int i=0; i < 100; i++)
    nums[i] = NULL;

for(int i=0; i < 100; i++) {
    int index = rand()%100;
    nums[index] = new int;
    *nums[index] = rand()%1000;
}

for(int i=0; i < 100; i++) {
    if(nums[i] != NULL) {
        cout << *nums[i] << endl;
    }
}
```

Problem (5) [10 points]

Assume there exists a “Rectangle” class with an outline as shown below. Add code so you can both cin and cout the dimensions of the rectangle. The cin format will be “NUMBERxNUMBER”, and the cout format will be “NUMBER by NUMBER” In other words, make it so main() and will cout “4 by 3 and 10 by 9” if an input of “4x3<enter>10x9<enter>” is given. (Note: in the input “<enter>” is the enter key being pressed).

```
class Rectangle {
private:
    double height;
    double width;
}

int main() {
    Rectangle x,y;
    cin >> x >> y;
    cout << x << " and " << y << endl;
}
```

Problem (6) [10 points]

Write (in C++) an appropriate destructor for the following constructor.

```
class leakyweaky {
public:
    int* x;
    double** y;
    double* z;
    leakyweaky(int a);
};

leakyweaky::leakyweaky(int a) {
    x = new int;
    *x = a;
    if(*x > 10) {
        y = new double**[*x*2];
        for(int i=0; i < *x*2; i++) {
            y[*x*2-i-1] = new double;
            z=y[i];
        }
    }
}
```

Problem (7) [10 points]

What is the output of this code?

```
class A {
public:
    virtual void v();
    double dub();
};

void A::v() {
    cout << "AV" << endl;
}

double A::dub() {
    v();
    return 8;
}

class B : public A {
public:
    void v();
    double dub();
};

void B::v() {
    cout << "BBVV" << endl;
}

double B::dub() {
    v();
    return 3;
}

int main() {
    A x;
    B y;
    A* z = &y;
    cout << x.dub() << endl;
    cout << y.dub() << endl;
    cout << z->dub() << endl;
}
```

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.

```
int* getx() {
    int x = 2;
    return &x;
}

int main() {
    int* mainX = getx();
    int* y = &mainX;
    cout << y* << endl;
    delete mainX;
}
```

Problem (9) [10 points]

What is the output of this code? You must show work for full credit.

```
string umm(int x) {
    if(x>1) {
        string you = umm(x/2);
        cout << "Are " << you << " ready?\n";
        return "yuppers";
    }
    else {
        return "bah";
    }
}

int main() {
    cout << "Bad " << umm(7) << endl;
}
```

Problem (10) [10 points]

Write code (in C++) that reads a single integer from the keyboard. This input number will dictate two things: the size of squares to be output and how many squares are output. The first square should be made of 'X's and then alternate between 'X's and 'O's. You may assume the input will be positive. See the examples below for more clarity on what is expected of the output:

Example 1, input of "3":

```
XXX000XXX  
XXX000XXX  
XXX000XXX
```

Example 2, input of "4":

```
XXXX0000XXXX0000  
XXXX0000XXXX0000  
XXXX0000XXXX0000  
XXXX0000XXXX0000
```

Problem (11) [10 points]

Write code (in C++) that will read a sentence from the keyboard. The code should count how many commas there are that are not inside quotes.

Example 1, input of: This, is, a, "weird," sentence.

Example 1, output: 3

Example 1, input of: "Quotes go until," the first "matching," end quote

Example 1, output: 0