

Multiple Choice Questions (circle ONE answer for each question)

[10 marks]

1. Given the program in Figure 1, which of lines (24) to (26) are valid and will not result in an error:
a. (24)
b. (25)
c. (26)
 d. (24) and (25)
2. Given the program in Figure 1, which of lines (28) to (30) are valid and will not result in an error:
 a. (28)
b. (28) and (29)
c. (30)
d. none
3. Given the program in Figure 1, which of lines (32) to (34) are valid and will not result in an error:
 a. (32)
b. (33)
c. (32) and (33)
d. (34)
4. Given the program in Figure 1, which of lines (36) to (38) are valid and will not result in an error:
a. (36)
b. (36) and (37)
 c. (38)
d. none
5. Given the program in Figure 1, which of lines (40) to (42) are valid and will not result in an error:
 a. (42)
 b. (40) and (41)
c. none
d. all of them

8. Given the class definition below:

```
class Tlist {  
    class Node {  
        friend Tlist;  
        private: Animal* data; Node* next;  
    };  
    public:  
        Tlist(); ~Tlist();  
        void pushBack(Animal*); void pushFront(Animal*);  
        Animal* popBack(); Animal* popFront();  
    private: Node* head; Node* tail;  
};
```

write the complete `pushFront()` function that adds the given element to the front of the list.

[10 marks]

```
void TList::PushFront(Animal* a) {
```

```
    Node* temp = new Node(); 2  
    temp->data = data; 1
```

```
    if (Head == NULL):
```

```
    {  
        Head = temp;
```

```
        temp->next = NULL; 1
```

```
    }
```

```
    else
```

```
    {
```

```
        Node* tempHead = head;
```

```
        head = temp;
```

```
        temp->next = tempHead; 3
```

```
    }
```

```
    return 0;
```

```
}
```

⑨

Programming Questions

7. Given the class definition below:

```
class Dlist {  
    class Node {  
        friend Dlist;  
        private: Animal* data; Node* prev; Node* next;  
    };  
public:  
    Dlist(); ~Dlist();  
    void pushBack(Animal*); void pushFront(Animal*);  
    Animal* popBack(); Animal* popFront();  
private: Node* head;
```

write the complete popBack() function that removes the last element at the back of the list and returns that element as the return value. Remember to manage your memory.

[10 marks]

Animal Dlist::popBack()

```
if (Head == NULL) return NULL;  
if (Head->next == NULL){  
    delete *head->data;  
    Animal* temp = Head->data;
```

[20 marks]
[ks]

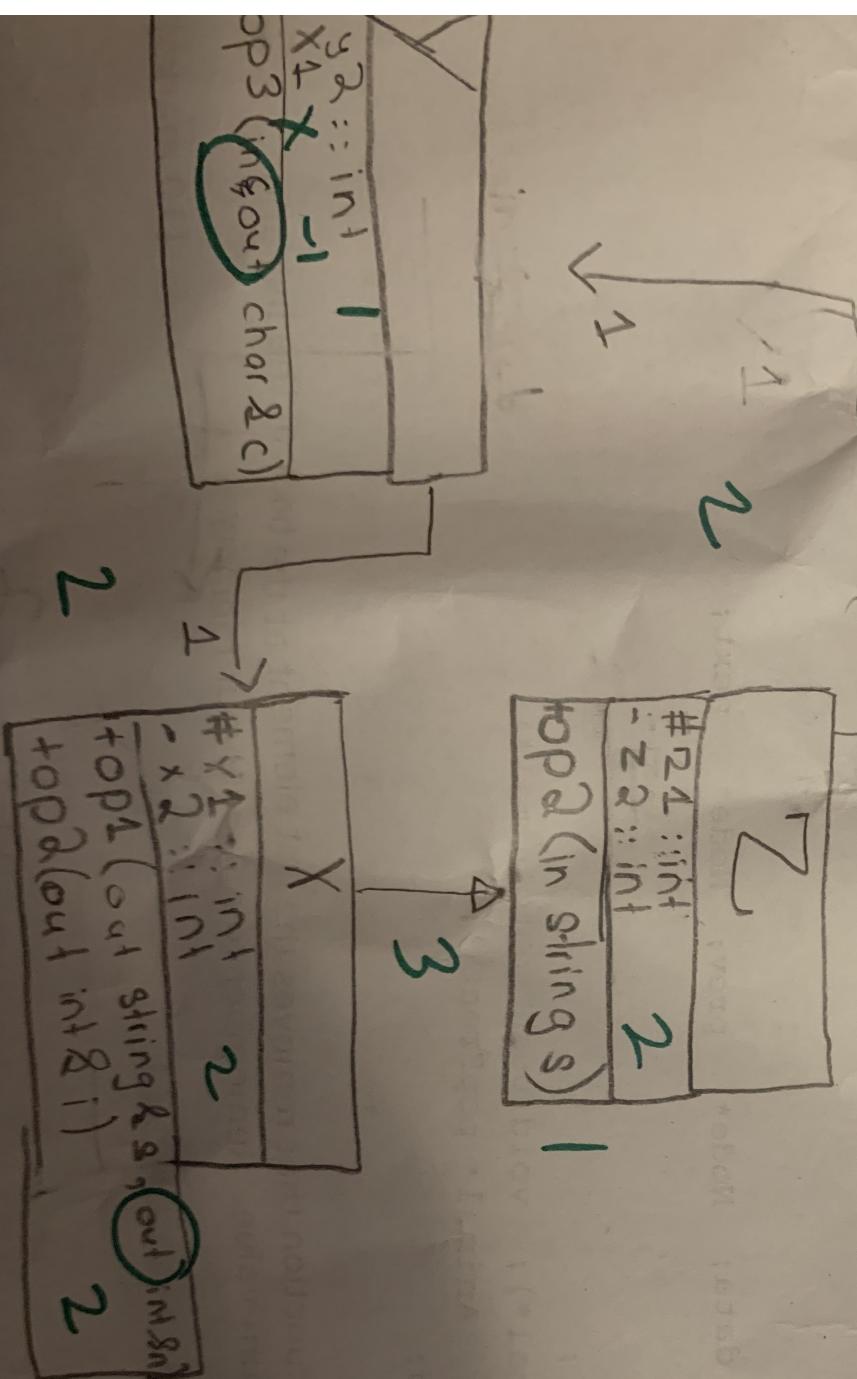
COMP 2404

Midterm Exam

3/50
out of 50
Authorized Memoranda:

UML Question

6. Draw the UML diagram corresponding to the classes defined in Figure 1. You must show all classes, and all attributes, operations, associations, directionality, and multiplicity, where applicable:



```
01 class Z {
02     public:    void op2(string s) { s = "xyz"; }
03             Y y1;
04     protected: int z1;
05     private:   int z2;  };
06
07 class X : public Z {
08     public:
09         void op1(string& s, int& n) { if (n > 0) s = "abc"; }
10         void op2(int& i) { i = 99; }
11     protected: int x1;
12     private:   int x2;  };
13
14 class Y {
15     public:    bool op3(int k, char& c) { if (k < 10 && c > 0) c = 'J'; }
16             int y1;
17     protected: int y2;
18     private:   X x1;  };
19
20 int main() {
21     X x;  Y y;  Z z;
22     string tmp = "hello";  int n1 = 12;  int n2 = 34;  char c = 'C';
23
24     x.op2(n1);
25     x.op2(tmp);
26     z.op2(n2);
27
28     x.z1 = 0;
29     x.z2 = 0;
30     z.op1(tmp,n1);
31
32     z.y1.op3(n1,c);
33     y.x1.op2(n2);
34     y.z1.op2(tmp);
35
36     y.x1.op2(n1);
37     y.x1.op2(tmp);
38     x.y1.op3(n1,c);
39
40     x.y1.y1 = 0;
41     x.y1.op3(n1,c);
42     y.x1.x1 = 0;
43
44     return 0;
45
46 }
```

Figure 1