CMPSC 24 Winter 2019 Midterm-I Answer Sheet

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

Q1(i)(10 pts): Define and implement the `Line` class
Parts(a)-(c)

```
class Line {

public:
    Line(double slope=0, double intercept=0):
        m(slope), c(intercept) { }
    double getSlope() const {return m;}
    double getIntercept() const {return c;}

private:
    const double m;
    const int c;

};
```

int is acceptable

int acceptable as well

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

Q1 (ii) (a) (5 pts) Implement the operator | |

```
bool operator|| (const Line & l1, const Line & l2){

    return  l1.getSlope() == l2.getSlope()

    // Strictly speaking students should check
    // for approximate equality using the fabs function
    // but this answer is acceptable
}
```

Q1 (ii) (b) (5 pts) Implement the operator−

```
Line operator- (const Line &l1, const Line & l2 ){
    // const member variables can only be initialized
    // via the constructor
    double a = l1.getSlope() - l2.getSlope();
    double b = l1.getIntercept() - l2.getIntercept();

    return Line(a,b);
}
```

CMPSC 24 Winter 2019 Midterm-I Answer Sheet

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

Q2(i) (8 pts)

(a) Compile-time error: Yes No

Output : 2 7

(b) Compile-time error: Yes No

Member variables are const.
Values cannot be reassigned using the assignment operator.

(c) Compile-time error: Yes No

y = 2*x+7
0x9000
y = 1*x +5

(d) Compile-time error: Yes No

Don't deduct points for formatting errors

y = 2x+7 y = 4x+5 intersect

y = 2*x+7 y = 2*x+10 don't intersect

(ii)(2 pts) Yes

-1*(l2 - l1).getIntercept() == -1*(l2.getIntercept() - l1.getIntercept())
== (l1.getIntercept() - l2.getIntercept())

Give full credit if answer is Yes

By the same logic
(l2 - l1).getSlope() == l2.getSlope() - l1.getSlope()

CMPSC 24 Winter 2019 Midterm-I Answer Sheet

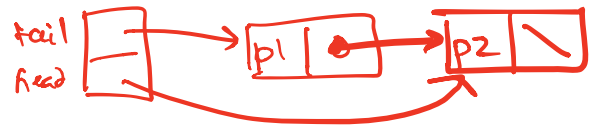
Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

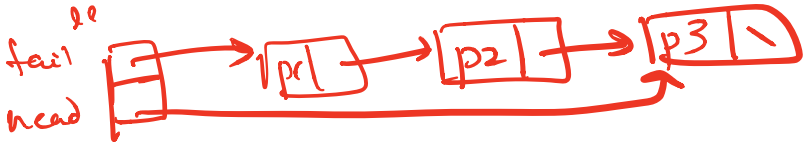
Q3 (i) (a) (2 pts) Select among options: A, B, C:
Justify your answer below: A

Original List

ll



List after append(p3)

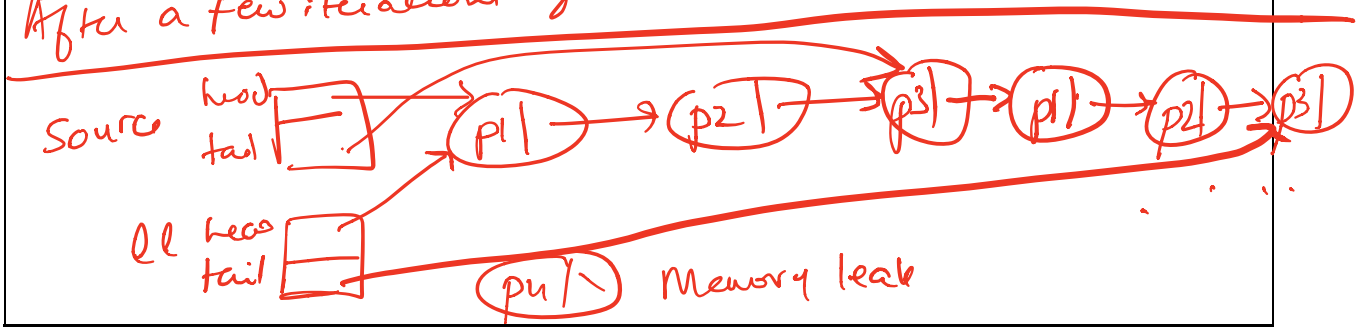


(b) (2 pts) Select among options: A, B, C
Justify your answer below: C (Infinite loop and memory leak)

Source

Original List

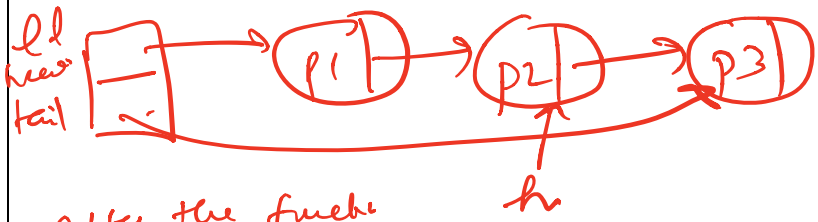
After a few iterations of the while loop



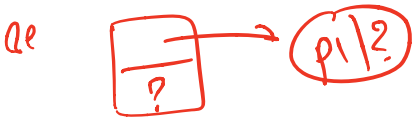
Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

(c) (2 pts) Select among options: A, B, C
Justify your answer below: A (Give partial credit if students chose C and gave the reason as head & tail are not correctly updated)



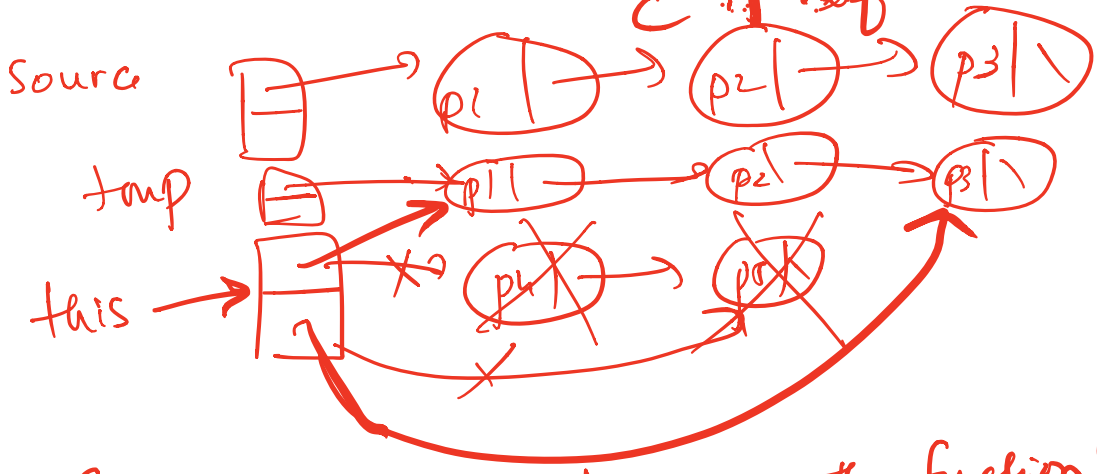
after the function



p2 & p3 are deleted
Okay because of pre-post conditions

A if default destructor is not correctly overloaded
C if default destructor is correctly overloaded

(d) (2 pts) Select among options: A, B, C
Justify your answer below:



So far its okay but after the function returns temp is deleted by the destructor. If the destructor was correctly implemented, all the nodes shared by temp & the linkedlist are also deleted.

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

(e) (2 pts) Select among options: A, B, C
Justify your answer below:

C, Students may have omitted this subtlety. So A is also acceptable

```
if (h->data == pname)
```

Incorrect type comparison
Only works if operator == was overloaded for Player type

Q3.b (ii) (5pts) Implement the function **`deleteMid()`**

```
void LinkedList::deleteMid(const string& pname){
    Node *prev, Node *curr = head;
    while (curr->data.name != pname)
    {   prev = curr;
        curr = curr->next;
    }
    prev->next = curr->next;
    delete curr;
}
```

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

Q3.b (iii) (5pts) Implement the function **`incrementScore()`**

```
/*Don't deduct points if students assumed that
a correct implementation of find is
available */

void LinkedList::incrementScore(const string & pname) {

Player p;
p.name = pname;
p.score = 1;
if (!find(pname))
    append(p);
    return;

}
```

CMPSC 24 Winter 2019 Midterm-I Answer Sheet

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____