MTH 4300/4299: Algorithms, Computers, and Programming II

Fall 2025

Midterm 2 Review

1 TRUE OR FALSE

1. Returning by reference allows you to avoid copying an object, improving performance, but returning by value ensures the caller receives a new copy of the object.
2. The **this** pointer in C++ is a constant pointer that holds the address of the current object.
3. The size of an std::vector is fixed once it is created and cannot change.
4. std::list in C++ allows random access to its elements, just like std::vector.
5. You can overload the arrow (->) operator in a class, and it is commonly used when a class contains or behaves like a pointer.
6. In a singly linked list, each node has a pointer to the previous node as well as the next node.
7. The time complexity of the selection sort algorithm is O(n log n) in the worst case.
8. The fstream class in C++ allows for both input and output operations on files.
9. Overloaded operators must have at least one operand that is of user-defined type(a class).
10. (*ptr).method() is the same as ptr->method()

2 SHORT ANSWER

1. How do you declare a member function that guarantees it will not modify the object it belongs to?
2. What happens to the elements of a vector when it resizes after exceeding its current capacity?
3. What is an advantage of using std::list over std::vector?
4. In stl the list stl data structure has a method named push_front() that adds an element to the front of the list. What is the time complexity of this method?
5. What operator do you have to overload as friend function(typically)?
6. What class in fstream is used to only open files?
7. When is a destructor called?
8. Given the files `main.cpp` `myclass1.cpp` `myclass1.h` how would you compile these in a terminal
9. Write any function signature that uses default parameters(arguments).
10. How does the selection sort algorithm determine which element to swap at each step?

3 CODING

1. Write the code for the method(adds a node to the end of a linked list. Return true if successful otherwise false. Do not use stl):

```
bool LinkedList::push_back(int val)
{
    ...
}
```

2. The code below has more than 5 errors. Find at least 5 for full credit!
rectangle.h:

```
#ifndef RECTANGLE_H
#define RECTANGLE_H

class Rectangle {
public:
    Rectangle(double width, double height);  // Constructor
    double getPerimeter() const;             // Member function to get perimeter
    void setHeight(double height) const;     // Sets height
    void getHeight();                        // Gets height


private:
    double width;                            // Member variables
    double height;
};

#endif // RECTANGLE_H
```

rectangle.cpp:

```
#include<iostream>
#include "rectangle.h"

// Constructor definition
Rectangle::Rectangle(double width, double height) : width(width), height(height) {}

// Function to calculate the area of the rectangle
double Rectangle::getArea() const
{
    return width * height;
}

// Sets height
double Rectangle::getPerimeter() const
{
    return 2 * (width + height);
}

// Function to calculate the perimeter of the rectangle
double Rectangle::setHeight(double h) const
{
    height=h;
}

// Gets height
void getHeight()
{
    return height;
}
```

main.cpp:

```
#include <iostream>

int main()
{
    Rectangle rect(10.0, 5.0);  // Create a Rectangle object with width 5.0 and height 3.0
    std::cout << "Area: " << rect.getArea() << std::endl;
    std::cout << "Perimeter: " << rect.getPerimeter() << std::endl;
    std::cout <<rect<< std::endl;
    return 0;
}
```

4 TRUE OR FALSE(SOLUTIONS)

1. T
2. T
3. F
4. F
5. T
6. F
7. F
8. T
9. T
10. T

5 SHORT ANSWER(SOLUTIONS)

1. void method() **const**{...}
2. They are copied over to another array that is twice as large.
3. constant time insertion and deletion at the beginning, no wasted memory by doubling size of vector, etc
4. O(1)
5. << or >>
6. ifstream
7. At the end of the scope (second "}" where caraible is defined)
8. g++ main.cpp myclass1.cpp
9. void func(int x=5,int y=7)
10. Finds the max at each step and places it towards the end of the data structure

6 CODING(SOLUTIONS)

1.

```
bool LinkedList::push_back(int val)
    {
        Node* newNode = new Node;
        newNode->data=val;
        newNode->next=vnullptr;

        if(!head)
        {
            head=newNode;
            return true;
        }

        Node* curr=head;
        while(curr->next) curr=curr->next;

        curr->next=newNode;
        return true;
    }
```

2.

1. missing #include<rectangle.h> in main.cpp
2. operator for insertion called for the rectangle class, but not overloaded
3. getArea() method not declared in #include<rectangle.h>
4. setHeight() method is declared as const but changes private attribute height
5. getArea() method does not return the correct type