# ECE 244F - Programming Fundamentals (Fall 2021)

## Midterm Test

Examiners: T.S. Abdelrahman, Mizan Rahman and Sadegh Yazdanshenas

**Duration: 110 minutes**

**This version of the midterm has typos fixed**

**Total Mark**

**Instructions – Read Carefully**

This test is OPEN Textbook and CLOSED notes. The use of computing and/or communicating devices is NOT permitted.

Do not remove any sheets from this test book. Answer all questions in the space provided. No additional sheets are permitted.

You may write your answers using a pencil or a pen.

Work independently. The value of each question is indicated. The total value of all questions is 100.

In answering the questions, you must assume the C++ 11 standard and the use of the g++ compiler available on the Linux machines in the ECF labs.

There may be some multiple-choice questions for which incorrect answers carry part marks. This is indicated in the relevant questions.

There may be some multiple-answer questions, where there is a penalty for incorrect answers. This is indicated in the relevant questions.

If you have any doubts about a question, write your assumption down. If they are sensible, they will be considered.

Best of luck!!

**Question (10 marks)**.*Warm Up Questions.*

Answer the following questions by circling the most appropriate answer.

1. cin is \_\_\_\_\_\_\_\_.
2. a class
3. an object
4. a package
5. a namespace
6. The this keyword in a method is \_\_\_\_\_\_\_\_\_\_.
7. the object on which a method is invoked
8. a pointer to the method that is invoked
9. a pointer to the object on which the method is invoked
10. a pointer to the class to which the method belongs
11. The copy constructor is executed when \_\_\_\_\_\_\_\_\_\_.
12. an object is assigned from another object of the same type
13. an object is created as a copy of another object of the same type
14. an object is passed to a function and the pass mechanism is by reference
15. b. and c.
16. None of the above
17. What does “#pragma once” in a header file do?
18. It instructs the preprocessor to only include the header file once
19. It instructs the g++ compiler to run once
20. It instructs the g++ compiler to run any function with a prototype in the header file only once
21. It instructs the operating system to run the entire program only once
22. The key advantage of separate compilation is \_\_\_\_\_\_\_\_.
23. to use more command to compile and link
24. to speed up the program development cycle
25. to discover more bugs
26. There is really no advantage
27. An object file (e.g., main.o) can be executed by changing its name to main.exe and typing the command ./main.exe at the Unix command prompt.

**True or False**

1. If you define your own copy constructor for a class, the compiler will not automatically generate a default constructor.

**True or False**

1. A constructor can be a private member of the class.

**True or False**

1. An auto variable can be defined without being initialized.

**True or False**

1. When cin encounters an error in the input stream, program execution is aborted.

**True or False**

**Question (2 marks)**.*Functions.*

Consider the following function:

void something (int a, int& b) {

 int c;

 c = a + 2;

 a = a \* 3;

 b = c + a;

}

What is the output of the following code fragment that invokes the function (assume iostream is included the std name space is used and that the code compiles and runs)?

int r = 1;

int s = 2;

int t = 3;

something(t, s);

cout << r << ' ' << s << ' ' << t << endl;

|  |  |
| --- | --- |
| **Answer:** |   |

**Question (4 marks)**.*Parameter Passing.*

The main function of a program declares and initializes a variable, loc as follows:

     int\* loc = new int;

The main function invokes a function called reallocate that has a ***return type of* int\*** and takes ***one formal argument***: a variable called the\_loc. The function is to allocate a new integer, and modifies its actual argument, loc, to store the address of the new integer and returns the old value of the argument. Which of the following are correct prototype and invocation of this function? ***Circle all correct answers***.

Note that this is a *multiple answers*question. The total number of points for the question are divided by the number of correct answers to the question. This amount is **awarded for every correct answer** selected and **is deducted for every incorrect answer** selected. Your mark cannot go below zero.

1. *Function prototype*: int\* reallocate (int\*\* the\_loc);

*Invocation in main*:  reallocate (&loc);

1. *Function prototype*: int\* reallocate (int\*& the\_loc);

*Invocation in main*:  reallocate (&loc);

1. *Function prototype*: int\* reallocate (int\* the\_loc);

*Invocation in main*:  reallocate (\*loc);

1. *Function prototype*: int\* reallocate (int\*& the\_loc);

*Invocation in main*:  reallocate (loc);

1. *Function prototype*: int\* reallocate (int\*& the\_loc);

*Invocation in main*:  reallocate (\*loc);

1. *Function prototype*: int\* reallocate (int\* the\_loc);

*Invocation in main*:  reallocate (\*loc);

**Question (5 marks)**.*Unix and Labs.*

* 1. **(2 marks)**. The following is a directory hierarchy on a Unix system, like the one used on ECF machines. The current working directory is /home/users/dce/ece244/lab1. Write a Unix command *that uses as few characters as possible* to change the current directory to /home/users/dce/ece244/lab2.

 

|  |  |
| --- | --- |
| **Answer:** |   |

* 1. **(1 mark)**. What is the name of the command used to change access permissions for your directories on ECF (just the name of the command)?

|  |  |
| --- | --- |
| **Answer:** |   |

1. **(1 mark)**. The exercise command compiles your source code, runs the resulting executable with some test cases and reports differences in output. **True or False** (Circle one answer)
2. **(1 mark)**. If a program has passed all exercise test cases for an assignment, you are guaranteed to receive full marks for the assignment. **True or False** (Circle one answer)

**Question (7 marks)**.*Compilation.*

Consider a program consisting of the following 6 files. Assume there are no errors in any files.

* Shapes.cpp is the implementation of member functions of a class named Shapes.
* Shapes.h is the header file containing the definition of the Shapes class.
* Student.cpp is the implementation of member functions of a class named Student.
* Student.h is the header file containing the definition of the Student class.
* globals.h is a header file containing some global definitions used in both Student and Shapes classes.
* Shapes.cpp includes Shapes.h and globals.h
* Student~~s~~.cpp includes Student.h and globals.h
* main.cpp includes Student.h, globals.h and Shapes.h. This file that contains the main function.
1. If we want to compile the program using a single command, what should that command be? Assume the executable is called main.exe. Write it below
2. If we instead want to use separate compilation, we require 4 commands. Write them below in the right order.
3. If we modify main.cpp, which of the commands in **(b)** do we need to rerun? Write them below in the right order.
4. If we modify Shapes.h, which of the commands in **(b)** do we need to rerun? Write them below in the right order.
5. If we modify globals.h, which of the commands in **(b)** do we need to rerun? Write them below in the right order.

**Question (4 marks)**.*Compilation.*

Consider the following two user-defined include files: file\_a.h and file\_b.h.

// File: file\_b.h

#ifndef file\_b\_h

#define file\_a\_h

// Contents

#endif

// File: file\_a.h

#ifndef file\_a\_h

#define file\_a\_h

// Contents

#endif

* 1. **(2 marks)**. Assume the main function of a program includes the two files as follows:

#include “file\_b.h”

#include “file\_a.h”

#include “file\_b.h”

#include “file\_a.h”

Which of the following statements is correct? Circle all answers that are correct but keep in mind that circling an incorrect answer carries a penalty.

* + 1. Each of the two include files is included only once
		2. The file file\_a.h is included but the file file\_b.h is not included
		3. The file file\_a.h is not included but the file file\_b.h is included
		4. The file file\_a.h is included twice
		5. The file file\_b.h is included twice
		6. Neither file is included
		7. Each of the two include files is included twice
	1. **(2 marks)**. Now assume that the main function includes the two files as follows:

#include “file\_a.h”

#include “file\_b.h”

#include “file\_a.h”

#include “file\_b.h”

Which of the following statements is correct? Circle all answers that are correct but keep in mind that circling an incorrect answer carries a penalty.

1. Each of the two include files is included only once
2. The file file\_a.h is included but the file file\_b.h is not included
3. The file file\_a.h is not included but the file file\_b.h is included
4. The file file\_a.h is included twice
5. The file file\_b.h is included twice
6. Neither file is included
7. Each of the files is included twice

**Question (5 marks)**.*Constructors and Destructors.*

Consider the class definition shown below. Assume that iostream and string are included, and that the std namespace is used. Also assume that the class is correctly implemented.

class Book {

 public:

 Book() {cout << “A new book is added.” << endl;}

 Book(string s) {

 cout << “Book “ << s << “ is added.” << endl;

 name = s;

 }

 ~Book() {cout << “Book “ << name

 << ” is removed.” << endl;}

 private:

 string name = “ECE244”;

};

What is the output of the following main function?

 Book \*b1;

int main() {

 Book b2;

 Book \*b3 = new Book(“Pride and Prejudice”);

 if (true) {

 b1 = new Book();

 Book b4(“Wuthering heights”);

 delete b3;

 Book \*b5 = new Book [2];

 }

 delete b1;

 return 0;

}

Write your answer in the table below, one line per table row. Please note that you may or may not need all the rows in the table.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Question (7 marks)**.*Classes.*

Consider the definition of a class called Mystery, which is in the file Mystery.h.

#include <iostream>

using namespace std;

class Mystery {

 private:

 // Private members not shown

 public:

 // Public members not shown

};

Now consider the following program that uses the Mystery class. The program compiles and runs correctly.

#include <iostream>

#include <string>

using namespace std;

#include “Mystery.h”

int main () {

 string n = “hello”;

 Mystery x(3,n);

 Mystery\* p = new Mystery(n,4);

 Mystery y(x);

 x.invert(1,n);

 if (x == y) x = y / \*p;

 delete p;

 return 0;

}

What public members of the class Mystery ***must exist*** (either written by the programmer or given by C++) for the above code ***to compile with no errors***? Non-member functions are not allowed. Give variable declarations and/or method prototypes in the table below. Note that you may or may not need to fill every row in the table.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Question (5 marks)**. *Pointers.*

Consider the code shown below. You can assume it compiles with no errors and runs.

#include <iostream>

using namespace std;

int a = 0;

int \*b = &a;

int \*\*c = &b;

int\* foo( int \*\*d ){

 (\*\*d)++;

 b = \*d;

 int \*e = new int;

 \*e = 10;

 return e;

}

int main(){

 int \*g = nullptr;

 int \*f = new int;

 \*f = 5;

 a++;

 // Point 1

 g = foo(&f);

 a++;

 (\*g)++;

 // Point 2

 return 0;

 }

In the table below, give the values of the variables indicated in the table columns when program execution reaches each of the two points, Point 1 and Point 2. If a value cannot be obtained due to dereferencing a nullptr pointer, write nullptr (but assume the program does not stop).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | a | \*b | \*\*c | \*g | \*f |
| Point 1 |  |  |  |  |  |
| Point 2 |  |  |  |  |  |

**Question (7 marks)**.*Pointers and Structures.*

Consider the following struct definition:

struct node{

 int id;

 node\* next;

 float\* valuep;

};

Now consider the following pointer variable declarations.

node\* head;

node\*\* p\_head = &head;

In the code segments that you write below, you are **not allowed** to create additional pointers and **must** perform all the tasks using p\_head. That is, you **cannot use** head.

* 1. **(3 marks)**. Write a code segment to dynamically allocate a node, make head point to it, assign its id field to 1 and dynamically allocate a float value in with valuep pointer. Keep in mind that the only variable you can use is p\_head. A schematic is shown.



Write your answer here

* 1. **(4 marks)**. Write a code segment to dynamically allocate a second node, assign its id field to 2, have the next filed of the first node point to this new node, then and dynamically allocate an array of size 2 floats with its valuep pointer. Keep in mind that the only variable you can use is p\_head. A schematic is shown.



Write your answer here

**Question (6 marks)**.*Dynamic Memory Allocation.*

We would like to create a *dynamically* allocated array called a, which has size n (the value of which is obtained from user input). Each of the n elements of the array points to a *dynamically* allocated array of exactly 4 elements. Each of these 4 elements points to a *dynamically* allocated array of 10 Shape objects.

Assume that the definition and implementation of the Shape class are included and are correct. Write a code snippet that allocates the data structure, as described above. ***Make sure to include all variable declarations***.

Write your answer here

**Question (5 marks)**. *Dynamic Memory Allocation.*

Study the following program and answer the questions below. You may assume that the class Mystery is correctly defined an implemented.

#include “Mystery.h”

void level\_B (Mystery obj) {

 Mystery\* ptr = new Mystery();

 // Point Z

 return;

}

void level\_A (Mystery\* x) {

 Mystery a;

 // Point Y

 Mystery\* ptr = new Mystery[10];

 delete [] x;

 // Point W

 return;

}

int main() {

 Mystery a;

 Mystery\* b;

 // Point X

 b = new Mystery[2];

 level\_A(b);

 level\_B(a);

 // Point Q

 return (0);

}

1. Indicate the number of objects of type Mystery that exist in memory when the program execution reaches Point X.

|  |  |
| --- | --- |
| **Answer:** |  |

1. Indicate the **change** in the number of objects of type Mystery that exist in memory and that occurs during program execution between Point X and Point Y. For example, if 5 more objects exist, write +5. If two fewer objects exist, write -2.

|  |  |
| --- | --- |
| **Answer:** |  |

1. Indicate the **change** in the number of objects of type Mystery that exist in memory and that occurs during program execution between Point Y and Point W. For example, if 5 more objects exist, write +5. If two fewer objects exist, write -2.

|  |  |
| --- | --- |
| **Answer:** |  |

1. Indicate the **change** in the number of objects of type Mystery that exist in memory and that occurs during program execution between Point W and Point Z. For example, if 5 more objects exist, write +5. If two fewer objects exist, write -2.

|  |  |
| --- | --- |
| **Answer:** |  |

1. Indicate the **change** in the number of objects of type Mystery that exist in memory and that occurs during program execution between Point Z and Point Q. For example, if 5 more objects exist, write +5. If two fewer objects exist, write -2.

|  |  |
| --- | --- |
| **Answer:** |  |

**Hint**: Draw a picture!

**Question (4 marks)**. *Functions and Objects.*

Consider a (non-member) function called doIt, which takes a single object of type DayOfYear and returns a single object also of type DayOfYear. You may assume that the class DayOfYear is correctly implemented and that DayOfYear.h is included. Which of the following implementations of this function is problem-free? Indicate your answer by placing an **X** in the appropriate column in the table.

|  |  |  |
| --- | --- | --- |
| Implementation | Problem-Free? | Has a problem? |
|  DayOfYear\* doIt(DayOfYear & arg) { DayOfYear temp; temp = arg; return (&temp); } |  |  |
|  DayOfYear doIt(DayOfYear & arg) { DayOfYear temp; temp = arg; return (temp); } |  |  |
|  DayOfYear & doIt(DayOfYear & arg) { DayOfYear temp; temp = arg; return (\*this); } |  |  |
|  DayOfYear doIt(DayOfYear & arg) { DayOfYear temp; temp = arg; return (\*this); } |  |  |

**Question (10 marks)**. *C++ I/O.*

1. **(4 marks)**. Write a C++ function void readInts() that repeatedly reads integers from the standard input (using cin) and then immediately outputs the input integer (using cout), one integer per line. When a ‘.’ character is encountered, the function prints the message Done on a line by itself and returns. You may assume that there will be no errors in the input integers.

Thus, for example if the user enters 5 6 7 ., the function prints:

5

6

7

Done

Write your code in the box below.

|  |
| --- |
| #include <iostream>using namespace std;void readInts() {} |

1. **(6 marks)**. Rewrite the function in part **(a)** to handle errors in the input. Thus, if the user enters any characters other than integer digits or the ‘.’, the function prints the message Error on a line by itself and returns.

Thus, if the user enters 1 2 3 a 4, the function prints:

1

2

3

Error

 Write your code in the box below. Assume no EOF will be given by the user.

|  |
| --- |
| #include <iostream>using namespace std;void readInts() {} |

**Question (4 marks)**.*Classes and Operator Overlaoding.*

The following is the definition/implementation of a class called Foo.

class Foo {

 private:

 int priv;

 public:

 void Foo (int pv) {priv = pv;}

 void Foo (const Foo\* src) {priv = src.priv;}

 Foo& operator=(Foo& rhs) {

 priv = rhs.priv;

 return (\*this);

 }

 int getPriv() const { return priv; }

 void setPriv(int pv) const { priv = pv; }

};

Compiling the above definition/implementation results in one or more errors. Re-write the class so it is error-free. Write your answer (the entire definition/implementation) in the box below. You may want to underline the changes you made in your answer below.

**Question (7 marks)**.*Operator Overloading*.

A vector is a mathematical representation of a physical quantity that has a magnitude and a direction. For example, a force has a magnitude (e.g., 2 newtons) and a direction (e.g., 45 degrees from the horizontal). The figure below shows a vector V and its magnitude |V| and direction . One way to represent vectors is using their components in the horizonal and vertical directions, as shows by Vx and Vy in the figure.



The following class is used to create objects that represent vectors using their components.

#include <iostream>

using namespace std;

class Vector {

 private:

 float Vx;

 float Vy;

 public:

 Vector(float x, float y);

 float getVx();

 float getVy();

 void setVx(float x);

 void setVy(float y);

 void print();

};

Vector::Vector(float x, float y) {

 Vx = x;

 Vy = y;

}

float Vector::getVx() {return (Vx);}

float Vector::getVy() {return (Vy);}

void Vector::setVx(float x) {Vx = x;}

void Vector::setVy(float y) {Vy = y;}

void Vector::print() {cout << Vx << “ ” << Vy << endl;}

We wish to overload the “+” operator for the Vector class to be able to add two objects of type Vector and write code like this in a non-member function (say main):

 Vector X(6,10);

 Vector Y(4,3);

 :

 .. = X + Y;

For example, if X represents a vector with Vx and Vy components of 6 and 10, and Y represents a vector with Vx and Vy components of 4 and 3, then X + Y results in a new object that represents a vector with Vx and Vy components of 10 and 13 (i.e., the sums of the respective Vx and Vy components).

Write the implementation of the overloaded operator function as a member of the class Vector. Clearly show the function header and its body.

Write your answer in the box below. Be sure to indicate both the header and the body of the method. You need not worry about using const modifiers.

{

Write function

header here

}

Write function

body here

**Question (8 marks)**.*Constructors.*

Consider the following definition for class Time and functions printHour and printMinute.

class Time{

 private:

 int hour, minute, second;

 public:

 Time(){hour = minute = second = 0;}

 Time(int h, int m, int s){hour = h; minute = m; second = s;}

 Time& operator=(const Time& rhs){

 hour = rhs.hour;

 minute = rhs.minute;

 second = rhs.second;

 return (\*this);

 }

 Time(const Time& rhs) {

 hour = rhs.hour;

 minute = rhs.minute;

 second = rhs.second;

 }

 int getHour() const { return hour; }

 int getMinute() const { return minute; }

};

void printHour(Time t){cout << t.getHour() << endl;}

void printMinute(Time& t){cout << t.getMinute() << endl;}

How many times are the assignment operator, default constructor, copy constructor, and the constructor with 3 integer parameters called for each expression? Assume the expressions are error-free and are written in the order shown in the table. The answers for the first two are given.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Expression | default constructor | copy constructor | assignment operator | Constructor with 3 ints  |
| Time a; | 1 | 0 | 0 | 0 |
| Time b(1,1,1); | 0 | 0 | 0 | 1 |
| Time arr1[10]; |  |  |  |  |
| Time e(); |  |  |  |  |
| Time\* arr2[20]; |  |  |  |  |
| Time c = a; |  |  |  |  |
| b = a; |  |  |  |  |
| arr2[19] = new Time(1,1,0); |  |  |  |  |
| printHour(b); |  |  |  |  |
| printMinute(b); |  |  |  |  |

**THIS PAGE IS BLANK FOR ANSWER OVERFLOW**