**CPSC 1020 Spring 2019**

**FINAL EXAM**

**This is a Closed Book exam. Please keep your notes and your computers closed.**

**Question 1(6 Points) C++ Polymorphism**

Review Attachment 1 and answer the following questions

1. When I run this program I get the following output.

1: I am a dog weighing 45.5 pounds.

2: I am a dog weighing 45.3 pounds.

3: I am a dog weighing 24.7 pounds.

However, I was expecting a different output (listed below):

1: I am a dog weighing 45.5 pounds.

2: I am a sheepdog weighing 45.3 pounds

and guarding 50 sheep.

3: I am a dog weighing 24.7 pounds.

1. Explain why the incorrect output is printing.
2. Changing only one line of code in this program can fix the above issue. Explain how can you change the program to make it print the correct output.
3. Line 44 (kennel[k]->bark()) is not a safe way to call the bark function.
   1. Why is this not safe?
   2. The vector class provides us a way to call the bark function that is safe. Below write the safe way to call the bark function.

**Question 2 (2 Points)C++ Memory Allocation, Structs**

struct Pixel{

unsigned char red, green, blue;

};

**Part 1:**

Assume Row and Column have been defined. In C++, dynamically allocate the memory for a **2D Array** of type Pixel called **pixArray.**

**Question 3 (2 Points) shared\_ptrs**

**Consider the following program. What is the output?**

#include<iostream>

#include<memory>

using namespace std;

OUTPUT:

int main(){

int\* s = new int;

shared\_ptr<int> s1(s);

cout << s1.use\_count() << endl;

shared\_ptr<int> s2 = s1;

shared\_ptr<int> s3(s2);

cout << "s1 count " << s1.use\_count() << endl;

//releasing the shared pointer s3

s3.reset();

cout << "s2 count " << s2.use\_count() << endl;

cout << "s3 count " << s3.use\_count() << endl;

s = nullptr;

return 0;

}

**Question 4: (3 Points) Reference variables**

What is the output of this program:

#include <iostream>

using namespace std;

int main()

{

OutPut:

double first = 15.3;

double second = 18.5;

double& ref\_d = first;

cout << first << endl;

cout << ref\_d << endl;

ref\_d = second;

cout << first << endl;

cout << ref\_d << endl;

second = 19.8;

cout << first << endl;

cout << ref\_d << endl;

return 0;

}

**Question 5 (5 Points) Member Initialization list**

**Consider the code on attachment 2 and complete the Person constructor using the member initialization list technique.**

Person::Person()

Person::Person(string l, string f, string e, int number, string street,string city, string state, int zip, int month, int day, int year)

**Question 6 (15 Points)**

Using attachment 3

This is the MyString class you wrote in lab.

For this question you will implement several of the functions from the MyString.h class. **You can assume the functions not listed here but listed in the MyString.h (attachment3) have already been implemente**d.

#include "MyString.h"

**/\*You must implement this constructor. You must use member initialization list**

**\*Hint: you will need to dynamically allocated memory and since this is a string**

**\*make sure you account for the null pointer. Also, you can call new in an initialization**

**\*list\*/**

MyString::MyString (const MyString& string)

{

}

**/\*Implement the destructor\*/**

MyString::~MyString ()

{

}

**/\*Implement this operator+. Pay close attention to the parameters and take**

**\*advantage of the constructors and operators that have already been implemented.**

**\*Using constructors and other operaors, this can be implemented in one line.\*/**

MyString operator+ (const MyString& left, const char\* right)

{

}

**/\* Implement this function.**

**\*This function provides bounds checking for when using [] \*/**

char MyString::operator[] (int ndx) const

{

}

**/\*You must implement this function. This function will output the string\*/**

ostream& operator<<(ostream& out, const MyString& string)

{

}

**Question 7 (2 Points)**

Consider the following program:

Change the given class to demonstrate how we can combine the two given constructors.

class Tank

{

private:

int gallons;

public:

Tank()

{

gallons = 50;

}

Tank(int gal)

{

gallons = gal;

}

int getGallons()

{

return gallons;

}

};

**Question 8 (5 Points) Function Pointer**

**Consider the following code. This is a “C” programming question:**

#include <stdio.h>

/\*functions all with the same signature\*/

void add(int, int);

void subtract(int, int);

void multiply(int, int);

int main()

{

int i;

**/\*1. Create an array of function pointers called funPtrs and point**

**it to the three functions implemented below(prototypes above).\*/**

unsigned int ch, a = 15, b = 10;

for(i = 0; i < 3; i++)

{

**/\*2. Write the code that calls the function pointer created above\*/**

}

return 0;

}

void add(int a, int b){

printf("Addition is %d\n", a+b);

}

void subtract(int a, int b){

printf("Subtraction is %d\n", a-b);

}

void multiply(int a, int b){

printf("Multiplication is %d\n", a\*b);

}

**Question 9 (6 Points)**

**Considering Attachment 4**:

The attachment 4 program was compiled using the -fno-elide-constructors flag. This flag ensures the compiler will not optimize (leave out) any of the constructor calls.

int main() {

MyClass tempClassObj; // Create object of type MyClass

MyClass anotherClassObj;

//make sure you understand what is called here

anotherClassObj = tempClassObj;

// Calls SomeFunction(), hint: tempClassObj is passed by value and has a return by value.

SomeFunction(tempClassObj);

return 0;

What is the output of main:

}

**Question 10 (4 points)**

What is the output of the following program? Show your work or your thought process.

#include <stdio.h>

int f(int , int \*, int \*\*);

Output:

int main(){

int c, \*b, \*\*a;

c = 4;

b = &c;

a = &b;

printf("%d \n", f(c, b, a));

return 0;

}

int f(int x, int \*py, int \*\*ppz){

int y, z;

\*\*ppz += 1;

z = \*\*ppz;

\*py += 2;

y = \*py;

x += 7;

return x + y + z;

}

**For the remaining questions circle the correct choice to question. Each worth .667 of a point.**

**Question 11**

When a virtual member function of a class in a class hierarchy is invoked through a pointer to an object in the class hierarchy, the compiler will select the member function to be invoked

A) from the class of the type of the pointer.

B) from the class of the object that is pointed to.

C) from the base class of the inheritance hierarchy.

D) from the derived class of the base pointer object.

E) None of the above

**Question 12**

True/False:

Pointers to a base class may be assigned the address of a derived class object.

**Question 13**

Polymorphism in C++ will not work unless \_\_\_\_\_\_\_\_.

A) pointers or references are being used

B) the members of the class are public

C) the <polymorphic> header file is included

D) All of the above

E) None of the above

**Question 14**

A pure virtual function

A) must be overridden in a derived class for the function to be useful.

B) executes more efficiently than a non-pure virtual function.

C) must be accompanied by a virtual constructor of the same class.

D) All of the above

E) None of the above

**Question 15**

When the body of a member function is defined inside a class declaration, it is called a(n) \_\_\_\_\_\_\_\_ function.

A) static

B) global

C) inline

D) conditional

E) constructor

**Question 16**

True/False:

A constructor is a public class function that is automatically invoked (i.e., called) whenever a class object is created.

**Question 17**

A constructor may have a return type of

A) int

B) bool

C) void

D) any of the above.

E) none of the above.

**Question 18**

True/False:

If employee is an array of objects with a public member function named setHourlyWage, the following statement correctly calls this method for employee[2].

employee.setHourlyWage[2](20.00);

**Question 19**

True/False:

The amount of memory used by an array depends upon the array's data type and how many elements it can hold.

**Question 20**

A pure virtual function

A) is a virtual function that has been completely debugged.

B) is a virtual function that has no implementation.

C) is a virtual function that performs a single task in program.

D) is one that is only called from within a virtual constructor.

E) None of the above

**Question 21**

You can assign the contents of one array to another by using

A) the assignment operator.

B) the equality operator.

C) both array names.

D) A and C together.

E) none of the above.

**Question 22**

Consider the following:

Assume we have a complete class called Date.

Assume we create the following:

1. Date first(10,7,1963);
2. Date second = first;

Line 2 of the above snippet of code will invoke which of the following:

Circle the correct answer.

a) copy constructor

b) assignment operator

**Question 23**

In C++, polymorphism is based on the ability to make member functions of a class

A) static.

B) protected.

C) virtual.

D) encapsulated.

E) None of the above

**Question 24**

True/False:

In C++ polymorphism is very difficult to achieve unless you also use inheritance.

**Question 25**

The \_\_\_\_\_\_\_\_, also known as the address operator, returns the memory address of a variable.

A) asterisk ( \* )

B) ampersand ( & )

C) percent sign (%)

D) exclamation point ( ! )

E) None of the above

**Question 26**

When you work with a dereferenced pointer, you are actually working with

A) a variable whose memory has been deallocated.

B) a copy of the value pointed to by the pointer variable.

C) the variable whose address is stored in the pointer variable.

D) All of the above

E) None of the above

**Question27**

The code segment int \*ptr; has the same meaning as

A) int ptr;.

B) \*int ptr;.

C) int ptr\*;.

D) int\* ptr;.

E) None of the above

**Question 28**

Polymorphism is when \_\_\_\_\_\_\_\_ in a class hierarchy perform differently, depending upon the class of the object making the call.

A) base class constructors

B) derived class destructors

C) member functions

D) derived class constructors

E) None of the above

**Question 29**

Which of the following statements will correctly carry out the operation stated in the comment to its right.

A) array 2 = array1 // Copy the elements of array 1 into array 2.

B) cout << array2 // Output the elements stored in array 2.

C) array2 = 5; // Place a 5 in each element of array2.

D) None of the above.

E) A and B, but not C.

**Question 30**

Object composition is useful for creating a \_\_\_\_\_\_\_\_ relationship between classes.

A) friend

B) static

C) has-a

D) conditional

E) None of the above

**Question 31**

When overloading the operator ++, \_\_\_\_\_\_\_\_ is used to distinguish preincrement from postincrement.

A) a parameterless data type

B) the keyword void

C) a dummy integer parameter

D) the placeholder [ ]

E) None of the above

**Question 32**

Protected members of a base class are like \_\_\_\_\_\_\_\_, with the exception that they may be accessed by derived classes.

A) constructor functions

B) static members

C) private members

D) public members

E) None of the above

**Question 33**

Which arithmetic operations can be performed on pointers?

A) All arithmetic operations that are legal in C++

B) Multiplication, division, addition, and subtraction

C) Addition , subtraction , preincrement, and postincrement

D) Only multiplication and addition

E) None of the above

**Question 34**

An abstract class is somewhat restricted in how it can be used because

A) it cannot use dynamic binding for its member functions.

B) it cannot use static binding for its member functions.

C) All of its members must be public.

D) the compiler does not allow objects of the class to be created.

E) None of the above

**Question 35**

A pointer may be initialized with

A) the address of an existing object of the appropriate type.

B) the value of a floating-point constant.

C) the value of a floating-point variable.

D) All of the above

E) None of the above

**Question 36**

The delete operator should only be used on pointers that

A) have not yet been used.

B) have been correctly initialized.

C) point to storage allocated by the new operator.

D) are appropriately dereferenced.

E) None of the above

**Question 37**

) If arr is an array identifier and k is an integer, the expression arr[k] is equivalent to

A) \*(arr + k).

B) \*arr + k.

C) &arr[k].

D) arr + k.

E) None of the above

**Question 38**

True/False:

The expression s->m has the same meaning as (\*s).m.

**Question 39**

Which of the following statements is not valid C++ code?

A) int ptr = &num1;

B) int ptr = int \*num1;

C) float num1 = &ptr2;

D) All of the above are valid.

E) All of the above are invalid.

**Question 40**

Suppose that a function dynamically allocates a block of memory with a local pointer variable p pointing to the allocated block. Suppose further that there are no other pointers referencing that block of memory, and the function returns without doing a delete on p. Then

A) the pointer p becomes a dangling pointer.

B) the compiler will automatically deallocate the memory pointed to by p.

C) the program will suffer from memory leaks.

D) the returning function will throw the bad\_alloc exception.

E) None of the above

**Question 41**

The set of operations supported by the unique\_ptr class include

A) the dereferencing operators \* and ->, the post and pre increment operators ++, and the post and pre decrement operators --.

B) the dereferencing operators \* and ->.

C) the assignment operator.

D) the delete operator.

E) None of the above

**Question 42**

The statement cout << &num1; will output

A) the value stored in the variable called num1.

B) the memory address of the variable called num1.

C) the number 1.

D) the string "&num1".

E) None of the above

**Question 43**

The statement cin >> \*p;

A) stores the keyboard input into the variable p.

B) stores the keyboard input into the pointer called p.

C) is illegal in C++.

D) stores the keyboard input into the variable pointed to by p.

E) None of the above

**Question 44**

A reason for passing a pointer to a function is

A) to avoid the overhead of copying large data structures.

B) to allow the called function to modify a variable accessible to the calling function.

C) to allow easy access to data in the function that is being called.

D) A and B are both true.

E) None of the above

**Question 45**

To add up all the values in a two-dimensional array it would be best to use

A) one for loop

B) two separate for loops

C) a nested for loop

D) no loop

E) one sentinel controlled loop

**Question 46**

A virtual function is declared by placing the keyword \_\_\_\_\_\_\_\_ in front of the return type in the base class's function declaration.

A) virtual

B) private

C) public

D) protected

E) None of the above

**Question 47**

True/False:

In C++ If you attempt to store more data in an array than it can hold, the compiler will issue an error.

**Question 48**

An abstract class is

A) one that has at least one pure virtual function.

B) one that supports polymorphism.

C) one that has all of its member functions declared private.

D) one that has all of its member variables declared private.

E) None of the above

**Question 49**

True/False: Object-oriented programming is centered around objects that include both data and the functions that operate on them.

**Question 50**

To indicate that a member function of a class is pure virtual,

A) you use the keywords pure virtual.

B) you must put = 0 where the body of the function would go.

C) you must include the <pure> header file.

D) All of the above

E) None of the above

**Question 51**

The process of having a class contain an instance of another class is known as

A) object overloading.

B) operator overloading.

C) object composition.

D) dynamic composition.

E) None of the above

**Question 52**

True/False: A class declaration provides a pattern for creating objects, but doesn’t make any objects.

**Question 53**

An object typically hides its data, but allows outside code to access it through its

A) private member functions.

B) public member functions.

C) public data members.

D) access specifiers.

E) None of the above

**Question 54**

It is a good idea to make a copy constructor's parameters \_\_\_\_\_\_\_\_ by specifying the \_\_\_\_\_\_\_\_ keyword in the parameter list.

A) inline, inline

B) static, static

C) constant, const

D) global, glob

E) None of the above

**Question 55**

\_\_\_\_\_\_\_\_ to a base class may be assigned the address of a derived class object.

A) Access specifiers

B) Static members

C) Private members

D) Pointers

E) None of the above

**Question 56**

\_\_\_\_\_\_\_\_ allows us to create new classes based on existing classes.

A) Polymorphism

B) Inheritance

C) Function overloading

D) The copy constructor

E) None of the above

**Question 57**

True/False:

A derived class may become a base class, if another class is derived from it.

**Question 58**

In the statement class Car:public Vehicle, which is the base class?

A) Car

B) Vehicle

C) public

D) class

E) None of the above

**Question 59**

The \_\_\_\_\_\_\_\_ class constructor is called before the \_\_\_\_\_\_\_\_ class constructor.

A) base, derived

B) derived, base

C) public, private

D) private, public

E) None of the above

**Question60**

If you do not furnish a(n) \_\_\_\_\_\_\_\_ for a class, a default will be provided for you by the compiler.

A) copy constructor

B) constructor

C) assignment operator

D) All of the above

E) None of the above

**Question 61**

True/False:

A member function of a derived class may not have the same name as a member function of a base class.

**Question 62**

True/False:

A derived class may not have any classes derived from it.

**Question 63**

If a member variable is declared \_\_\_\_\_\_\_\_, all objects of that class share access to that variable.

A) static

B) dynamic

C) inline

D) default

E) None of the above

**Question 64**

A good reason for overloading an operator is to enable it to

A) outperform its C language counterparts.

B) be used with types defined by the programmer.

C) operate on more operands than in its standard definition.

D) operate on no operands.

E) None of the above

**Question 65**

A member function that is declared \_\_\_\_\_\_\_\_ cannot use the this pointer.

A) private

B) public

C) static

D) inline

E) None of the above

**Question 66**

A \_\_\_\_\_\_\_\_ function is not a member of a class, but it has access to the private members of the class.

A) static

B) constructor

C) destructor

D) friend

E) None of the above

**Question 67**

A destructor is a member function that

A) is used to remove old unneeded objects.

B) causes the program to end.

C) is automatically called when an object is destroyed.

D) can only be called by the main function of a program.

E) None of the above.

**Question 68**

A(n) \_\_\_\_\_\_\_\_ member function may be called by a statement in a function that is outside of the class.

A) inline

B) public

C) private

D) declared

E) constructor

**Question 69**

C++ allows you to overload

A) compiler errors.

B) preprocessor directives.

C) operators and functions.

D) undefined variables.

E) None of the above

**Question 70**

The name of a destructor must begin with

A) the name of the class.

B) a tilde (~).

C) a capital letter.

D) an underscore.

E) none of the above.

**Question 71**

A(n) \_\_\_\_\_\_\_\_ is a special function that is called whenever a new object is created and initialized with data from another object of the same class.

A) destructor

B) static function

C) copy constructor

D) assignment function

E) None of the above

**Question 72**

When a member function is defined outside of the class declaration, the function name must be qualified with the class name, followed by

A) a semicolon(;).

B) the scope resolution operator (::).

C) the public access specifier.

D) the private access specifier.

E) a tilde (~).

**Question 73**

Each object of a class has its own copy of the class's

A) static member variables.

B) instance member variables.

C) static member functions.

D) All of the above

E) None of the above

**Question 74**

A(n) \_\_\_\_\_\_\_\_ member variable may be accessed before any objects of the class have been declared.

A) private

B) public

C) inline

D) static

E) None of the above

**Question 75**

True/False: By default, when an object is assigned to another, each member of one object is copied to its counterpart in the other object.