



Review Test Submission: Midterm 3

User	Stephen Giang
Course	PROGRAMMING_LANGUAGES
Test	Midterm 3
Started	11/18/20 4:00 PM
Submitted	11/18/20 5:09 PM
Due Date	11/18/20 5:15 PM
Status	Completed
Attempt Score	92 out of 100 points
Time Elapsed	1 hour, 8 minutes out of 1 hour and 15 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

0 out of 2 points

A base class object cannot be assigned to a derived class variable because:

Selected Answer: Objects are encapsulated.

Answers: The object has a different type.

There would be no members to assign to the derived class's added members.

Objects are encapsulated.

A derived class object can only be assigned to a derived class variable.

Question 2

2 out of 2 points

A makefile is a mechanism that:

Selected Answer: provides software project management.

Answers: provides software project management.

can be used with C++ only.

generates documentation.

is seldom used with multiple files.

Question 3

2 out of 2 points

A protected member of a class is accessible to its own class and to any members in a derived class.

Selected Answer: True

Answers: True

False

Question 4

2 out of 2 points

A struct in C++ has the same structure as a class except that it uses the keyword struct instead of class and:

Selected Answer: its member data are public by default.

Answers: its member data are public by default.

it must be used in a header file only.

only one struct is allowed per implementation file.

the name of the file it is declared in must match the name of the struct.

Question 5

2 out of 2 points

A subprogram is one of two fundamental abstraction facilities in programming languages. It is an example of :

Selected Answer: process abstraction.

Answers: data abstraction.

data distraction.

process inspection.

process abstraction.

Question 6

2 out of 2 points

An advantage of data abstraction is:

Selected Answer: Program organization, modifiability, and separate compilation.

Answers: Program organization, modifiability, and separate compilation.

Making actual definition details available to clients.

Eliminating the need for the compiler process.

Representing objects and their operations in multiple syntactic units.

Question 7

2 out of 2 points

Using C's signal.h header would provide the code necessary to enable the programmer to write code to pass a function pointer to another function when an event is created by an external action (such as the user hitting CTRL-C to halt a running program).

Selected Answer: True

Answers: True

False

Question 8

2 out of 2 points

Another term for dynamic binding is:

Selected Answer: late binding.

Answers: latent binding.

early binding.

static binding.

 late binding.

Question 9

0 out of 2 points

Assuming the appropriate include files and function definition, what is output from the following C++ code segment?

```
for(int i=0; i <= 6; i+=2)
    for(int j=0; j < 2; j+=2)
        cout << "x";
        cout << endl;
```

Selected Answer:  x
 x
 x
 x

Answers:  x
 x
 x
 x

 x
 x
 x
 x
 x
 x

xxxxxxxx

 xxxx

none of these

Question 10

0 out of 2 points

Bjorne Stroustrup first called C++ "C with glasses."

Selected Answer:  True

Answers: True
 False

Question 11

2 out of 2 points

C and C++ use header files for interfaces to library files.

Selected Answer:  True

Answers:  True
False

Question 12

0 out of 2 points

Class definitions in C begin with the keyword:

Selected Answer: class

Answers: begin

class

main

new

None of these

Question 13

2 out of 2 points

Determine the output of the following program.

```
#include <iostream>
using namespace std;
int operate (int a, int b) {
    return (a * b);
}
float operate (float a, float b) {
    return (a/b);
}
int main() {
    int x=5, y=2;
    float n=5.0, m=2.0;
    cout << operate(x,y) <<"\t";
    cout << operate (n,m);
    return 0;
}
```

Selected Answer: 10 2.5

Answers: 10.0 5.0

5.0 2.5

10 2.5

10.0 5

Question 14

0 out of 2 points

Encapsulation in Ada is accomplished by the mechanism of (the):

Selected Answer: class.

Answers: class.

package.

template.

dynamic allocation.

Question 15

0 out of 2 points

For object-oriented languages, which of the following is not a design issue?

Selected Answer: Are subclasses subtypes (e.g., derived classes)?

Answers: Are methods required to return only objects?

Can variables have the same name in different namespaces?

Single vs. multiple inheritance

Are subclasses subtypes (e.g., derived classes)?

Question 16

2 out of 2 points

Given a virtual (not pure virtual) function in a base class, it may be overridden by a virtual function in a derived class that has the:

Selected Answer: same name and same type and same number of formal parameters.

Answers: same name.

same name and same number of formal parameters.

same name and same type of formal parameters.

same name and same type and same number of formal parameters.

Question 17

2 out of 2 points

Given the C++ class MyClass, the class destructor prototype is:

Selected Answer: ~MyClass();

Answers: ~MyClass();

dtor();

end();

terminator();

Question 18

0 out of 2 points

Given the below C++ code, the variable my_float is _____.

```
class foo {  
    int my_number;  
    my_float;  
public:  
    foo(int n, float f);  
};
```

Selected Answer: A and C

Answers: a private data member of class foo.

a friend data member of the client.

a public data member of instances of class foo.

A and C

Question 19

2 out of 2 points

Given the output formatting statements in the segment of C++ code below, what will be the output?

```
int x = 1234;  
cout.setf(ios::right);  
cout.fill('.'); // change the fill character  
cout.width(10); // set field width to 10  
cout << x;
```

Selected Answer:1234

Answers: 1234.0

1234.....

.....1234

.....1234

Question 20

2 out of 2 points

In C++, a base class definition a method is defined as

```
virtual void printClass() = 0;
```

and the method declared in a class derived from that base class is:

```
void printClass() { cout << "Derived" << endl; }
```

The method printClass() in the base class is a _____.

Selected Answer: pure virtual function

Answers: virtual function

pure virtual function

virtual function and a base class constructor

virtual function and a derived class constructor

Question 21

0 out of 2 points

In C++, if in the base class definition a method is defined as

```
virtual void printClass() = 0;
```

then the method declared as

```
void printClass() { cout << "Derived" << endl; }
```

in a class derived from that base class, is a(n):

Selected Answer: overloaded function.

Answers: virtual function.

pure virtual function.

base class constructor.

overloaded function.

Question 22

2 out of 2 points

Large program encapsulation in C++ does not include use of:

Selected Answer: assemblies.

Answers: classes.

friends.

namespaces.

Question 23

2 out of 2 points

Multiple inheritance means

Selected Answer:  A subclass has more than one parent class.

Answers: A child class has more than one derived class.

Multiple classes can inherit a base class.

Two parent classes are virtual classes.

 A subclass has more than one parent class.

Question 24

2 out of 2 points

Nearly all programming languages support process abstraction with:

Selected Answer:  subprograms

Answers:  subprograms

classes

structs

abstract data types

Question 25

2 out of 2 points

One major advantage of object-oriented code is that:

Selected Answer:  it makes reusing code easier.

Answers: it allows new programs to begin from scratch.

it separates code and data.

 it makes reusing code easier.

it exposes data to outside changes.

Question 26

2 out of 2 points

Overloaded functions are:

Selected Answer:  Two or more functions with the same name but different number of parameters or type.

Answers: Very long functions than can hardly run.

One function containing one or more nested functions inside it.

 Two or more functions with the same name but different number of parameters or type.

None of the above.

Question 27

0 out of 2 points

In the code segment below, the default constructor is _____,

```
class foo {  
    int my_number;  
    float my_float;  
public:  
    foo(int n, float f);  
};
```

Selected Answer: exists and is called by default.

Answers: non-existent.

exists and is called by default.

exists but must be called explicitly.

exists and is called whenever the constructor foo that takes two parameters called.

Question 28

2 out of 2 points

Support for abstract data types in Java are user-defined data types called:

Selected Answer: classes.

Answers: enclosures.

packages.

classes.

elements.

Question 29

2 out of 2 points

Supposing that the following is taken from a (valid) makefile that contains source files game1.c, card.c, and deck.c, which files if modified would cause the executable file to be recompiled when make is invoked?

```
game1: game1.o card.o deck.o  
 ${CC} -o game1 game1.o card.o deck.o
```

Selected Answer: game1.c, card.c, deck.c, and any of their #include'd .h files

Answers: game1.o

game1.c, card.c, deck.c

game1.c, card.c, deck.c, and any of their #include'd .h files

game1, game1.o, card.o, deck.o

Question 30

2 out of 2 points

The (technical) difference between a pointer and a reference type in C++ is that a pointer refers to an address in memory, while a reference refers to:

Selected Answer: an object or a value in memory.

- Answers:
- an object or a value in memory.
 - a memory address.
 - a non-constant pointer.
 - a class definition.

Question 31

2 out of 2 points

The difference between pass by value and pass by reference as it relates to subprograms is that:

- Selected Answer: pass by reference can change the value of the passed parameter in the calling function.
- Answers:
- pass by value can change the address of the passed parameter in the calling function.
 - pass by reference can change the value of the passed parameter in the calling function.
 - pass by value can change the value of the passed parameter in the calling function.
 - pass by reference can change the address of the passed parameter in the calling function.

Question 32

2 out of 2 points

The following C++ code segment from a class definition shows an example of an inline method.

```
class B1 {  
    int b;  
public:  
    B1() { cout << "B1::B1()" << endl; }  
  
    B1(int i) : b(i) { cout << "B1::B1(int)" << endl; } // inline ctor  
};
```

The rationale for this design choice was to:

- Selected Answer: Increase run-time efficiency by eliminating the usual call and return from the method.
- Answers:
- Complicate the user interface.
 - Increase readability.
 - Increase the compile-time efficiency by requiring no optimization.
 - Increase run-time efficiency by eliminating the usual call and return from the method.

Question 33

2 out of 2 points

The list of parameters given in subprogram call statements are actual parameters.

- Selected Answer: True
- Answers:
- True
 - False

Question 34

2 out of 2 points

The problem with the (legal) assignment of a derived class object to a base class variable is:

- Selected Answer: It discards the parts of the derived class object that are not members of the base class.
- Answers:
- The object has a different type.
 - A derived class object must be assigned to one of its child class objects.

It discards the parts of the derived class object that are not members of the base class.

A base class object can only be assigned to a derived class variable.

Question 35

2 out of 2 points

The scope resolution operator in C++ is represented by the symbol:

Selected Answer: ::

Answers:

::

->

::/

Question 36

2 out of 2 points

What danger that exists in C++ is avoided in Java and C# by having implicit garbage collection?

Selected Answer:

Implicit garbage collection removes the necessity of allowing users to explicitly deallocate objects, thereby eliminating the possibility of user-created dangling pointers.

Answers: Allocating objects from the heap.

Implicit garbage collection removes the necessity of allowing users to explicitly deallocate objects, thereby eliminating the possibility of user-created dangling pointers.

Allowing the user to create variables that perform dynamic allocation.

None.

Question 37

2 out of 2 points

What is the only function all C++ programs must contain?

Selected Answer: main()

Answers: start()

system()

main()

program()

Question 38

2 out of 2 points

What must all object-oriented languages contain?

Selected Answer: abstract data types, inheritance, and polymorphism

Answers: abstract data types, inheritance, and polymorphism

inherent data types, inheritance and polymorphism

encapsulation, inheritance and metamorphic types

encapsulation, inherent data types and dynamic binding

Question 39

2 out of 2 points

What purpose does const have in the following C++ function definition?

```
void foo(const MyObject&){ ... }
```

Selected Answer: It prevents the method foo() from changing the value of MyObject.

Answers: It indicates that foo() does not return a value.

It prevents the method foo() from changing the value of MyObject.

It makes MyObject into a local pointer.

All of the above.

Question 40

2 out of 2 points

Which language does not support an object-oriented approach?

Selected Answer: Fortran90

Answers: Ruby

Smalltalk

C#

Fortran90

Question 41

2 out of 2 points

Which of the following access modifiers are defined in C# but not in C++?

Selected Answer: internal

Answers: private

protected

public

internal

Question 42

2 out of 2 points

Which of the following is a danger inherent in C's approach to encapsulation?

Selected Answer:

The user can copy the header file content into the application file which can lead to using subsequently updated application files without using the potentially updated header file.

Answers:

The user can copy the header file content into the application file which can lead to using subsequently updated application files without using the potentially updated header file.

C classes are declared virtual by the compiler.

It causes data abstraction to supersede process abstraction.

The linker may link duplicate copies of the same object file.

Question 43

2 out of 2 points

Given the following code for function `foo` and the call to it in `main`, what are the values of `val1` and `val2` if the operands in the expressions are evaluated left to right?

```
int foo(int *k) {  
    *k += 4;  
    return 3 * (*k) - 1;  
}  
...  
void main() {  
    int i = 10, j = 10, val1, val2;  
    val1 = (i / 2) + foo(&i);  
    val2 = foo(&j) + (j / 2);  
}
```

Selected Answer: None of these.

Answers: val1 is 48; and val2 is 46

val1 is 41; and val2 is 61

val1 is 46; and val2 is 48

None of these.

Question 44

2 out of 2 points

Where are all Java methods declared?

Selected Answer: Within a class definition.

Answers: In header files.

Within a class definition.

In private member functions.

In public member functions.

Question 45

4 out of 4 points

Briefly describe short-circuit evaluation and give an example of its use.

Selected Answer: Short-circuit evaluation is a way for the program to not have to evaluate all the operations within the line/expression.

Example:

`bool ex = (0 == 1 && 4 == 4);`

In this example, because `0 == 1` is false, it would make the entire variable equal to false, without even evaluating whether or not `4 == 4`.

Correct Answer: [None]

Question 46

4 out of 4 points

Do you think the elimination of overloaded operators in your favorite language would be beneficial? Why or why not?

Selected Answer: If we suppose that my favorite language is c++, then I would say that the elimination of overloaded operators would not be beneficial. I for one very much find them useful. The disadvantages that come with it is that it could reduce readability, or might make error checking harder. However the benefits of overloading operators would allow real world scenarios in programming. For example, we could say parent1 + parent2 = child with an overloaded operator. Without overloaded operators, it would cause a lot of calls to a specific method, and make the code less readable. To be honest, it really comes down to preference as long as the overloaded operators make sense logically, and all error checking is complete.

Correct Answer: [None]

Question 47

4 out of 4 points

Write a brief reflection regarding writing a Fortran program: what was interesting? What is one thing you can appreciate about the language?

Selected Answer: One thing that was super interesting to me was that Fortran doesn't allow you to have more than 130 characters in a single line of code. When I was coding the Fortran program, my style is to use a lot of indents and whitespace to format my code to be easier to read to me. I think I was writing a print statement and it exceeded the 130 character allowance, and it gave me an ArrayOutOfBounds error. I had to look it up because I wasn't using any arrays.

One thing I can appreciate about the language is that it's math based and math orientated. As a math major, finding languages that can help me evaluate and understand mathematical models are very helpful to me. That's why one of my favorite languages is Matlab.

Correct Answer: [None]

Question 48

10 out of 10 points (Extra Credit)

Write one sentence that describes your thoughts on online learning this year. Credit given for a complete sentence that is actually descriptive.

Selected Answer: Online learning has been insanely rough because I learn best when I can hear the opinions, questions, and thoughts of my fellow peers throughout the learning process.

Correct Answer: [None]

Sunday, December 13, 2020 6:51:27 PM PST

← OK