Review Test Submission: Midterm 2

User	Stephen Giang
Course	PROGRAMMING_LANGUAGES
Test	Midterm 2
Started	10/21/20 4:00 PM
Submitted	10/21/20 5:16 PM
Status	Completed
Attempt Score	85 out of 100 points
Time Elapsed	1 hour, 16 minutes out of 1 hour and 30 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

3 out of 3 points

Bindings created in a subroutine are destroyed when the subroutine exits.

Selected Answer: True
Answers: True
False

Question 2

3 out of 3 points

Some languages, allow keywords to be identifiers, as shown in this PL/1 statement.

```
IF ELSE = THEN THEN ELSE = IF; ELSE THEN = IF;
```

Selected Answer: True
Answers: True
False

Question 3

3 out of 3 points

The following code fragment is a valid C language statement.

```
char *s = {"Hello World"};
```

Selected Answer: False
Answers: True
False

Question 4

3 out of 3 points

Show the representation of -11 in two's complement form. Do not just give the representation. SHOW how it is

determined.

Selected Answer: Notice that 11 can be written as 1(2^0) + 1(2^1) + 0(2^2) + 1(2^3). So we get our bit representation of 11 is 00001011. As -11 we need to turn the first bit into a 1 to represent the negative sign. So we get -11 is 10001011.
To get twos complement form, we flip the bits. so we get 01110100. Now we add one: 00000001. The result is 01110101. Now we can convert this back into decimal form so we get: 1(2^0) + 0(2^1) + 1(2^2) + 0(2^3) + 1(2^4) + 1(2^5) + 1(2^6) + 0(2^7). That is 1 + 4 + 16 + 32 + 64 = 117.

Correct Answer:
00001011
11110100
\+ 1
✓ 11110101

Question 5

3 out of 3 points

A text editor written in C stores the text of a page in the following array:

```
char buffer[1000][80] = {'\0'};
```

Following this declaration, 100 lines of text have been entered by the user and stored sequentially in the buffer.

Write a single line of C language code that will read a line of text from the keyboard to replace what was previously read in on the 5th line.

Selected Answer: gets(buffer[4])

Correct Answer:
scanf("%s", buffer[4]);
fscanf(stdin, "%s", buffer[4]);
fgets(buffer[4], 80, stdin);
✓ gets(buffer[4]);

Question 6

0 out of 3 points

The following code fragment is a valid C language statement.

```
int *n;
n = malloc(sizeof(int));
*n = 2;
```

Selected Answer: ✗ False

Answers:
✓ True
False

Question 7

12 out of 12 points

REFLECTIVE ESSAY

Provide a summary of your coding experience with C and Python

This question has no expressly right or wrong answer. You will be graded on how completely you cover the required components; but do make your points succinctly.

Your answer should include:

-- How did you initially react to your first exposure to the language?

-- What was easy to grasp? What gave you difficulty and why?

-- What value do you see for future use of each language?

-- What are two similarities in the languages and what are two differences?

Selected Answer:	Expose to C was definitely very new to me. Understanding the malloc, pointers, and the indexing was definitely a tough one for me. Im so used to working with programming languages like python, Java, and C++, where it throws an IndexoutofBounds error, and having objects. My first exposure to Python was last year and that was definitely different coming from java. It was different because of the implicit typing. I thought that was so weird and that it uses english words like in and has dynamic lists already built in. In C, the easiest thing to grasp was the argc and argv. That just seemed to make sense, as the main function at the end of the day is still a function that needs parameters. The hardest thing was pointers, because we use the same syntax * to reference and dereference the pointer, and how the pointer can act like an array. In python, the easiest thing to grasp was the syntax. I dont think I have any difficulty in Python. The value I see in C are programs that definitely dont need OOP. I can see C being very primitive language so companies who havent made the shift to a language like C++. C also have very foundational things that show true skill of a programmer. For Python, it shows simply more modern syntax and being able to program given the size of an array, having a list, tuple, and a lot of things not given in C. Two similarities that they share are that they both are able to do pretty much do the same thing, like arrays, loops, and functions. They also share variable types such as integers, char, and booleans. Two differences that they have are that python doesn't use pointers and Python is interpreted whereas C is compiled.
Correct Answer:	[None]

Question 8

3 out of 3 points

In C, where `A` is declared as `char A[100]`, the notation `A[51]` is equivalent to `*(*A + 51)`.

Selected Answer:	✓ False
Answers:	True
	✓ False

Question 9

3 out of 3 points

In C, given that C stores arrays in row-major order and an array `data[2][5]` has been declared, the sixth array element stored is accessed by `data[1][1]`.

Selected Answer:	✓ False
Answers:	True
	✓ False

Question 10

3 out of 3 points

A type binding is an association between a name and its scope.

Selected Answer:	✓ False
Answers:	True
	✓ False

Question 11

4 out of 4 points

Given the C program below, select a set of values that will complete the blanks 1-4, so that the program outputs the values stored in array A in **column-major order.**

```
char A[3][4];
char c = 'A';
int i, j, k;

 for (i = 0; i < 3; i++) {      // rows
     for (j = 0; j < 4; j++ ) { // columns
        A[i][j] = c++;
     }
  }
// Print array 4 columns wide
  k = 0;
 for (i = 0; i < __BLANK1__; i++) {
     for (j = 0; j < __BLANK2__; j++ ) {
        printf("%c ", A[ _BLANK3_ ][ _BLANK4_ ]);
        if (++k % 4 == 0)
            printf("\n");
     }
  }
}
```

Selected Answer: Blank1: 4 Blank2: 3 Blank3: j Blank4: i

Answers: Blank1: 4 Blank2: 3 Blank3: j Blank4: i

Blank1: 3 Blank2: 4 Blank3: j Blank4: i

Blank1: 3 Blank2: 4 Blank3: i Blank4: j

Blank1: 4 Blank2: 3 Blank3: i Blank4: j

Question 12

0 out of 3 points

True or False? The following code fragment is a valid C language statement.

```
int n = 2, *p;
p = &n;
```

Selected Answer: False

Answers: True

False

Question 13

3 out of 3 points

In C, a `struct` stores different data types in the same memory location, while a `union` stores its data in different memory locations.

Selected Answer: False

Answers: True

False

Question 14

3 out of 3 points

Give the output of the following Python code segment. If the code contains an error, print "No output".

```
# Python3 program
def func( list ):

     first = list.pop( 0 )
     last = list.pop( - 1 )

     list.insert( 0 , last)
     list.append(first)

     return list

# Driver code
newList  = [ 12 ,  35 ,  9 ,  56 ,  24 ]

print(func(newList))
```

Selected Answer: [24 , 35, 9, 56, 12]

Correct Answer:

```
[24, 35, 9, 56, 12]
```

Question 15

3 out of 3 points

Give the number of times the following Fortran 90 loop will iterate.

```
INTEGER :: I
DO I=1, 25, 5
     PRINT *,  "I know Fortran!"
END DO
```

Selected Answer: 5

Correct Answer: [None]

Question 16

0 out of 3 points

The following is valid syntax for a Python function definition:

```
def aFunc(a, b)
   return a * b
```

Selected Answer: True

Answers: True

False

Question 17

3 out of 3 points

If an object outlives its binding, it is garbage.

Selected Answer: True
Answers: True
False

Question 18

3 out of 3 points

A heap-dynamic array has the advantage of flexibility: growing and shrinking as necessary during program execution.

Selected Answer: True
Answers: True
False

Question 19

3 out of 3 points

If a string data type is a *non-primitive* data type, then it may consist of an array of characters.

Selected Answer: True
Answers: True
False

Question 20

3 out of 3 points

A union is one way in which an alias is formed.

Selected Answer: True
Answers: True
False

Question 21

5 out of 5 points

Match each term with its definition.

Question	Correct Match	Selected Match
Lifetime	B. The period of time a variable is bound to a specific memory location.	B. The period of time a variable is bound to a specific memory location.
Variable scope	A. The textual region of a program in which a binding is active.	A. The textual region of a program in which a binding is active.
Binding time	D. The time at which an association between an attribute and an entity (such as between a variable and its type) takes place.	D. The time at which an association between an attribute and an entity (such as between a variable and its type) takes place.
Elaboration	H. The creation of bindings when entering a scope.	H. The creation of bindings when entering a scope.
Referencing environment	F. The set of active bindings.	F. The set of active bindings.

All Answer Choices

A. The textual region of a program in which a binding is active.

B. The period of time a variable is bound to a specific memory location.

C. Implicit variable

D.
The time at which an association between an attribute and an entity (such as between a variable and its type) takes place.

E. Type equivalence

F. The set of active bindings.

G. Lexical Scope rules

H. The creation of bindings when entering a scope.

I. Named constant

Question 22

3 out of 3 points

Fortran array subscript syntax improves readability.

Selected Answer: ✓ False

Answers: True

✓ False

Question 23

4 out of 4 points

Briefly describe the advantages and disadvantages of: --will accept advantages only because of error on test.

1) a static array

2) a fixed stack-dynamic array

Selected Answer:

The Advantages of a static array:
We get better efficiency with a static array as the size doesn't change so we wont need to allocate more and more space. We allocate a single memory space for the static array.
The Disadvantages of a static array:
A disadvantage of a static array is that for large programs, many static arrays take up to much memory.

The Advantages of a fixed stack-dynamic array:

We are able to share the same memory allocation as other fixed stack-dynamic arrays, as long as they are not active at the same time. This works because the array gets allocated when needed, and deallocated when no longer needed. This deallocation allows for the arrays to share the same memory allocation when one is in use and other is not.

The Disadvantages of a fixed stack-dynamic array:

The disadvantage being the amount of time needed to keep reallocating and deallocating the arrays.

Correct Answer:

✓
Static array: Advantage: execution efficiency (no allocation/deallocation needed); disadvantage: need to know size at compile time, bindings are fixed for the entire program execution.
Fixed stack-dynamic array: advantage: space efficiency, supports recursion. disadvantage: size must be known at compile time.

Question 24

0 out of 3 points

The following code fragment is a valid C language statement.

```
int n = malloc(2*sizeof(int));
int *p;
p = &n;
*(p+1) = 2;
```

Selected Answer: False
Answers: True
False

Question 25

3 out of 3 points

Python is an interpreted language.

Selected Answer: True
Answers: True
False

Question 26

3 out of 3 points

The following is a valid Python syntax for a loop:

```
num=3
while True:
    if (num % 10 == 0):
        break
    print(num)
```

Selected Answer: True
Answers: True
False

Question 27

0 out of 3 points

Which of the following is a valid argument against representing Boolean data types as single bits in memory?

Selected Answer: Some languages store "True" or "False" instead of a zero or one.

Answers: Some languages store "True" or "False" instead of a zero or one.

Many machines cannot efficiently access a single bit of memory.

Boolean values are best stored as binary encoded decimals.

Boolean values are used less frequently than other primitive types and do not require a separate data type.

Question 28

3 out of 3 points

Early binding times are generally more efficient, while late binding times are generally more flexible.

Selected Answer: True
Answers: True
False

Question 29

3 out of 3 points

An example of an ordinal type is the enum type in C.

Selected Answer: True
Answers: True
False

Sunday, December 13, 2020 6:50:21 PM PST

← OK