

Exam II

Due Nov 25 at 11:59pm **Points** 60 **Questions** 33
Available after Nov 21 at 9am **Time Limit** 50 Minutes

Instructions

This exam is open notes, open calculator, open compiler, and open internet. The exam is due by Sunday, November 25th, at 11:59pm. If you have not finished taking the exam by that time Canvas may not let you submit the test. Please plan your schedule accordingly.

The exam time limit is 50 minutes. You should set aside sufficient time to focus on the work and avoid distractions.

Human collaboration is strictly prohibited. You may not ask friends for help, copy code from others, or have any human assistance during the exam. If the instructor finds evidence that you cheated, you will receive a 0 grade on this exam.

You should expect that your code will reviewed and inspected for signs of plagiarism.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	50 minutes	49 out of 60

Score for this quiz: **49** out of 60

Submitted Nov 24 at 3:02pm

This attempt took 50 minutes.

Correct!

Question 1

1 / 1 pts

"cout" is able to display both C++ string objects as well as C-style strings.

☒ True

☐ False

Question 2**1 / 1 pts**

Array indices in C++ always start from 0.

Correct!☒ True☐ False**Question 3****1 / 1 pts**

C++ references can only be passed into a function and cannot be used as a return type.

Correct!☐ True☒ False**Question 4****1 / 1 pts**

Any iterative algorithm can be implemented in a recursive fashion.

Correct!☒ True☐ False**Question 5****1 / 1 pts**

In C++ the name of a 1-dimensional array can be used within the source code as a pointer to the first element within the array.

Correct!☒ True☐ False**Question 6****1 / 1 pts**

Accessing an out-of-range array index in C++ will be detected at compile time and your program will not compile.

☐ True**Correct!**☒ False**Question 7****1 / 1 pts**

If you have the following line of code:

```
char my_string[6] = {'C', 'S', '1', '6', '1'};
```

you can use **my_string[0]** to access the first character contained in the my_string array.

Correct!☒ True☐ False**Question 8****0 / 1 pts**

A C++ string object needs to have a NULL character to properly terminate the string.

You Answered

☒ True

Correct Answer

☐ False**Question 9**

1 / 1 pts

C++ pointers can only be passed into a function and cannot be used as a return type.

☐ True

Correct!

☒ False**Question 10**

1 / 1 pts

Variables allocated on the stack are said to be dynamically allocated.

☐ True

Correct!

☒ False**Question 11**

1 / 1 pts

In C++ it's considered good practice to set all unused pointers to a value of 1.

☐ True

Correct!

☒ False

Question 12**1 / 1 pts**

When a function calls itself, this behavior is known as recursion.

Correct!☒ True☐ False**Question 13****1 / 1 pts**

A void pointer is any pointer that has a NULL value.

Correct!☐ True☒ False**Question 14****1 / 1 pts**

In C++ the * operator is used strictly for integer and floating point multiplication.

Correct!☐ True☒ False**Question 15****1 / 1 pts**

A pointer to a pointer cannot exist in C++.

Correct!☐ True☒ False**Question 16****1 / 1 pts**

Dynamic memory allocation occurs when memory is allocated at compile-time.

☐ True**Correct!**☒ False**Question 17****2 / 2 pts**

Consider the following code snippet:

```
int my_array[12];  
my_array[0] = 14;  
int* var = &my_array[0];
```

After execution, **var** contains:

☐ The value 12☐ The value 14☐ The memory address of the last element contained in my_array.**Correct!**☒ The memory address of the first element contained in my_array.

Question 18**2 / 2 pts**

Write a declaration for a pointer named p1 that is capable of pointing to a **double**.

Correct!

```
double * p1;
```

Correct Answers

```
double* p1;
```

```
double * p1;
```

```
double *p1;
```

Question 19**0 / 2 pts**

Suppose a fellow student has implemented the game Yahtzee as part of their homework assignment. While playing the game, you notice that the values of the dice seem to be random for the first roll, but subsequent rolls always generate the same 5 values. What would be the likely cause?

☐ The dice array was not set with initial values.

Correct Answer☐

The student used rand() while initializing the dice array but forget to use it after each roll.

☐ The array of dice was declared with type double rather than type int.

You Answered

☒ The random number generator was not properly seeded.

Question 20**2 / 2 pts**

A variable that holds the address of another variable is said to be a:

Correct!

- ☐ reference
- ☐ redirector
- ☐ alias
- ☒ pointer

Question 21**2 / 2 pts**

If you want to create a C-style string, how many characters must you reserve for the NULL terminator?

Correct!

- ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4

Question 22**2 / 2 pts**

If you have a C++ string named **my_string** how could you determine the number of characters in the string?

Correct!

- ☐ getLength(my_string)
- ☐ length(my_string)
- ☐ my_string.getLength()
- ☒ my_string.length()

Question 23**0 / 2 pts**

Which of the following declarations is invalid?

☐ `float* fptr = NULL;`☐ `char* chptr = NULL;`☐ `char** ptr = NULL;`☒ `string* strptr = NULL;`

You Answered

Correct Answer

☐ Each of the declarations above are valid.**Question 24****2 / 2 pts**

Which of the following would declare a C++ reference to a **double**?

☐ `my_ref = double&;`☒ `double& my_ref;`☐ `my_ref = double*;`☐ `double* my_ref;`☐ `double** my_ref;`

Correct!

Question 25**2 / 2 pts**

Consider the following code snippet:

```
int my_array[12];  
my_array[1] = 14;  
int* var = &my_array[0];  
int* addr = var;
```

Which of the following statements is true?

- ☐ addr is a pointer to the memory address of the variable var.
- ☐ var contains the value of the first element within my_array.
- ☒ addr points to the first element within my_array.
- ☐ The first element of my_array contains the value 14.

Correct!

Question 26

2 / 2 pts

Suppose you declare the following:

```
double inside = 2.6;  
double* ptr = &inside;
```

Which of the following statements is not allowed?

- ☐ (*ptr) = 13.5;
- ☐ ptr = NULL;
- ☐ inside += 13.4;
- ☒ *ptr = &inside;
- ☐ cout << ptr << " " << *ptr;

Correct!

Question 27**2 / 2 pts**

A poorly written C++ program may dynamically allocate memory on the heap without remembering to deallocate it. In CS161 what is the term that we use to describe this problem? (two words)

Correct!

memory leak

Correct Answers

memory leak

Question 28**3 / 3 pts**

Assuming an otherwise functional program, what is the output of this code after execution?

```
void fun(int &x, int y, int *z) {  
    x++;  
    y++;  
    (*z)++;  
}  
  
int main() {  
    int i = 3, j = 2, k = 1;  
    fun(i, j, &k);  
    cout << "i is " << i;  
    cout << " j is " << j;  
    cout << " k is " << k << endl;  
    return 0;  
}
```

☐ i is 4 j is 3 k is 2☐ i is 3 j is 2 k is 1☐ i is 4 j is 2 k is 1**Correct!**☒ i is 4 j is 2 k is 2☐ i is 3 j is 3 k is 2

Question 29**0 / 3 pts**

In an otherwise functional program, explain why the following snippet of text is not valid C++ code. Assume that all other areas of the program are written correctly.

```
char my_char;  
float f = 17.8;  
float& my_ref;  
my_ref = f;
```

Your Answer:

Trying to allocate a variable type of float to a char reference.

Question 30**2 / 2 pts**

Consider the following code snippet:

```
int matrix[3][4];  
int k = 0;  
for(int i =0; i <= 2; i++)  
    for (int j=0; j <= 3; j++)  
        matrix[i][j] = j;
```

After execution, the value of matrix[2][3] is:

☐ 1

☐ 2

☒ 3

☐ 4

☐ Undefined

Correct!

Question 31**2 / 2 pts**

A recursive function that continually calls itself without a base case will cause what behavior?

- ☐ Generate an error at compile time.
- ☐ Operate correctly but will require nearly twice the execution time.
- ☐ Be unable to use any memory from the heap.
- ☒ Continue to use stack memory until all available memory is exhausted.

Correct!**Question 32****2 / 2 pts**

Consider the following code snippet:

```
int* ar;  
int size = 5;  
int k = 4;  
ar = new int[size];  
for (int i=0; i<size; i++) {  
    ar[i] = i*k;  
}
```

After execution, what is the value of ar[4] ?

Correct!**Correct Answers**

16.0 (with margin: 0.0)

Question 33**7 / 10 pts**

Write C++ code to declare and initialize a dynamically allocated 2-dimensional array of **unsigned integers**. All array elements must be located on the heap. The array should have 2 rows with 10 columns inside each row (i.e. the entire 2D array will contain 20 unsigned integers).

Using loops of your choice, write code that will insert 10 even unsigned integers (beginning with 30, all the way up to 48) into the first row of the array.

In the second row of the array, use a loop to insert 10 odd unsigned integers (beginning with 31, all the way up to 49).

After your code has fully executed, the array contents should mimic the following diagram:

30, 32, 34, 36, 38, 40, 42, 44, 46, 48

31, 33, 35, 37, 39, 41, 43, 45, 47, 49

Your code must also free (deallocate) the dynamic memory that you used.

Your code does not need to include headers, function definitions or anything besides the lines that are specifically necessary to fulfill the requirements given above. In other words, you can just write a code snippet with the relevant lines. You still need to use proper C++ syntax and formatting.

Your Answer:

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <climits>
#include <ctime>
#include <cmath>
#include <cstdlib>
#include <bits/stdc++.h>

using namespace std;

int main () {
    int result = 2;
    int ** scoresheet;
    scoresheet = new int* [result];
    for(int i=0; i<result; i++) {
        scoresheet[i] = new int[10];
    }
```

```
for (int col=0; col<10; ++col){  
    scoresheet[0][col] = (col*2)+30;  
}  
  
for (int col=0; col<10; col++){  
    scoresheet[1][col] = 30+(col*2+1)  
}  
  
}
```

Quiz Score: **49** out of 60