CS 161 Winter 2020 Practice Questions for Midterm 2

Enter your name, ID number, and form number (1 or 2) on the Scantron; leave the section number blank.

Use a #2 pencil to fill in the Scantron.

***** There are 110 points you can earn on this exam, scored out of 100. *****

Part I: True (T) / False (F) (20 pts, 2 pts each)

1. True. You can have the following two function prototypes in the same program:
```
void fun(int a, int b, int c);
int* fun(int a, int b);
```

2. False. Default arguments can be located anywhere in the parameter list of a function.

3. True. It is legal to subtract an integer from a pointer variable (e.g., `p - 3` if p is a pointer).

4. True. The name of an array stores the value of the first array element.

5. True. Assuming **`arr`** is an array of integers, and **`idx`** is an int variable that is less than the array length, then both of the following statements do the same thing.
```
cout << arr[idx] << endl;
cout << *(arr + idx) << endl;
```

6. False. With C++ reference variables, you can access and modify data in other variables, and you can change where these reference variables refer to at any point.

7. False. You may use the **+** operator to append two C-style strings.

8. False. C++ performs array bounds checking, making it impossible for you to assign a pointer the address of an element out of the boundaries of an array.

9. False. Suppose **`arr_2d`** is a 5 (rows) by 4 (columns) integer array, then elements **`arr_2d[1][1]`** and **`arr_2d[2][1]`** have contiguous memory addresses, meaning they reside next to each other in memory.

10. False. When deleting a dynamic 2D array on the heap, you first delete the row pointers and then the columns for each row pointer.

Part II: Multiple Choice (60 pts, 3 pts each)

1. D. Which of the following statements is not valid C++ code?
   A. **`int ptr = &num1;`**
   B. **`char arr[];`**
   C. **`float& r;`**
   D. None of these are valid.

2. A. An array can store a group of values, but the values must be:
   A. the same data type

B. each of a different data type
C. constants
D. integers

3. C. When you pass the name of an array to a function, the function receives__________.
A. the length of the array
B. a copy of the array
C. the address of the array
D. a copy of the value of the first element

4. A. Suppose you declare the following:
```
double radius = 5.0, pi = 3.14;
double *ptr = &pi;
```
Which of the following statements is not allowed?
A. `*ptr = &radius;`
B. `cout << *ptr;`
C. `(*ptr)++;`
D. `*ptr = 0;`

5. B. Which of the following function declaration can be passed the following array?
```
char myArray[3][4];
```
A. `void fun(char a[][], int size);`
B. `void fun(char a[][4], int size);`
C. `void fun(char [3][], int size);`
D. `void fun(char [][]a, int size);`

6. C. To assign the contents of one array to another, you must use __________.
A. the assignment operator (=) with the array names
B. the equality operator (= =) with the array names
C. a loop to assign the elements of one array to the other array
D. None of these

7. C. What will the following code output?
```
int year = 2020;
int *ptr = &year;
cout << ptr << endl;
```
A. 2020
B. A * followed by 2020
C. The address of the **year** variable
D. A * followed by the address of the **year** variable
E. The address of the **ptr** variable

8. A. Assuming **ptr** is a pointer variable to an int, what will the following statement output?
```
cout << *ptr << endl;
```
A. The value stored in the variable whose address is contained in **ptr**.
B. The string "***ptr**".
C. The address of the variable stored in **ptr**
D. The address of the variable whose address is stored in **ptr**.

9. C. Look at the following code:

```
int numbers[5] = {0, 1, 2, 3, 4};
int *ptr = numbers;
ptr++;
```

After this code executes, which of the following statements is true?

A. `ptr` will hold the address of `numbers[0]`.
B. `ptr` will hold the value of `numbers[0]`.
C. `ptr` will hold the address of `numbers[1]`.
D. `ptr` will hold the value of `numbers[1]`.
E. This code will not compile.

10. D. What will the following code do?

```
int SIZE = 5;
double array[SIZE];
for (int i = 1; i <= SIZE; i++)
     array[i] = 0.5;
```

A. Each element in the array is initialized to 0.5
B. Each element in the array, except the first, is initialized to 0.5
C. Each element in the array, except the first and last, is initialized to 0.5
D. This code has an error that may cause it to crash

Questions 11-14 are based on the following code:

```
void read_strings(____③_____ , int size) {
     for (int i = 0; i < size; i++)
          cin >> array[i];
}
int main(){
     string *arr = new string [5];
     read_strings(___①___ , 5); // Function call
     ________②_________ // Free allocated memory
     return 0;

}
```

11. B. Which of the following is valid at ①?

A. `arr[5]`
B. `arr`
C. `*arr`
D. `&arr`

12. C. Which of the following will correctly free memory at ②?

A. `delete arr;`
B. `delete arr [];`
C. `delete [] arr;`
D. `delete *arr;`

13. D. Which of the following is valid at ③?

A. `string *array;`
B. `string &array;`
C. `string array[];`
D. `A and C;`

14. A. Assume ① - ③ are correct, is each element in **`arr`** filled after calling **`read_strings()`**?

A. Yes. The memory address of the array has been passed into the function, and therefore it can change the content in the original array.
B. No. The function call passed a copy of the array into **`read_strings()`**, so the content in the original array does not change.

Questions 15-18 are based on the following code:

```
void print_arr( ____①____  , int a, int b) {
     // code
}
int main() {
     int **two_d_arr = new int*[3];
     for (int i=0; i < 3; i++){
          two_d_arr[i] = new int[4];
          for (int j=0; j < 4; j++)
               two_d_arr[i][j] = 1;
     }
     print_arr(___②____  , 3, 4);
     ________③_________  // Free memory
     return 0;
}
```

15. A. How many rows and columns does **`two_d_arr`** have, based on the code above?

A. 3 rows, 4 columns
B. 4 rows, 3 columns
C. Cannot be defined
D. None of the above

16. C. According to the code above, which of the following is correct?

A. The double pointer **`two_d_arr`**, the row pointers, and the row arrays are all on the stack
B. The double pointer **`two_d_arr`** and row pointers are on the stack, and the row arrays are on the heap
C. The double pointer **`two_d_arr`** is on the stack, and the row pointers and row arrays are on the heap
D. The double pointer **`two_d_arr`**, the row pointers, and the row arrays are all on the heap

17. A. Which of the following is valid for ① and ②?

A. ① `int** array`   ② `two_d_arr`
B. ① `int** array`   ② `&two_d_arr`
C. ① `int array[][]`   ② `two_d_arr`
D. ① `int array[3][4]` ② `two_d_arr`

18. A. Which of the following will correctly free memory at ③?

A.
```
for (int i = 0; i < 3; i++)
     delete [] two_d_arr[i];
delete [] two_d_arr;
```
B.
```
for (int i = 0; i < 4; i++)
     delete [] two_d_arr[i];
delete [] two_d_arr;
```
C. `delete [] two_d_arr;`

D. `delete [][] two_d_arr;`

19. C. What is the output of the following code, given the function definitions below?

```
void tester (int a, int &b){
     int c = 0;
     c = a + 2;
     a = a * 3;
     b = c + a;
}
int main () {
     int x = 2, y = 3;
     tester(y, x);
     cout << x << " " << y << endl;
     return 0;
}
```

A. 2 3
B. 2 10
C. 14 3
D. 14 9

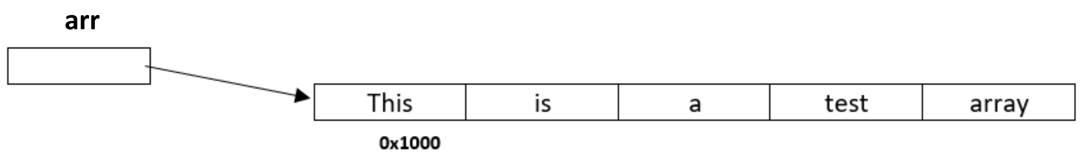
20. C. What does the following statement do?

```
int *ptr = NULL;
```

A. Create a variable named **`*ptr`** that will store an integer value.
B. Create a variable named **`*ptr`** that will store an asterisk (*) and an integer value.
C. Create a pointer variable named **`ptr`** that will store the address of an integer variable.
D. Create a variable named **`*ptr`** that will store the **`NULL`** value.

Part III: Short Answer (30 pts)

1. (10 pts, 2 pts each) This is the graphic interpretation of a **1D static array of C++ strings**, answer the following questions using the information on the graph.



a. What is the size (length) of the array?
5

b. Where is this array located at, stack or heap?
Stack (static array).

c. What is the value of **`arr[4]`**?
array

d. What is the content stored in **`arr`**?
0x1000 (address of first item)

2. (8 pts, 4 pts each) Assume the code fragment is embedded in an otherwise correct and complete program. Trace through the code, and write your answer in blank space.

   1) What is the output of the following code?

```
void my_fun(int x, int& y, int* z) {
     x = y + 3;
     y = x + *z;
     *z = y + x;
}

int main() {
     int a = 1, b = 2, c = 3;
     my_fun(a, b, &c);
     cout << a << "  " << b << "  " << c << endl;
     return 0;
}
```

```
1 8 13
```

   2) What is the value of **`matrix[3][4]`** after running the code below?

```
     int matrix [10][10];
     for (int i = 0; i < 10; i++)
          for (int j = 0; j < 10; j++)
               matrix[i][j] = (i+1) * (j+1);
```

20

3. (12 pts, 2 pts each) Suppose your program contains these statements:

```
string str = "kick";
string* p1 = &str;
string** p2 = &p1;
```

What is the type of:

`str`	`&str`	`*p1`
`string`	`string*` (not a reference)	`string`
`p2`	`&p2`	`*p2`
`string**`	`string***` (not a reference)	`string*`