CSC 150

Pre Lab #9: Arrays

Purpose:

1. Initializing and accessing array contents
2. Inputting, processing, and outputting array contents
3. Demonstration of accessing past array bounds
4. Passing arrays as arguments to functions

1. Complete the section exercises in chapter 10 of the course's textbook.

   Section 10.2 exercises (pages 277-278)
   Section 10.3 exercises (pages 281-282)
   Section 10.4 exercises (page 284)
   Section 10.6 (pages 291-292) (function declaration == prototype)

For the following problems, first attempt to determine their output by analysis of the code. You can then copy the code and paste it into a project to check your answer.

2. What does the following program print?

```
#include <iostream>
using namespace std;
int main()
{
  int i;
  int my_array[10];  // change this line in problem 3

  for (i = 0; i< 10; i++)
    my_array[i] = i;
  for (i = 0; i < 10; i++)
    cout << my_array[i] << "  ";
  return 0;
}
```

answer: 0 1 2 3 4 5 6 7 8 9

3. What happens if you change `int my_array[10];` to `int my_array;`

answer: 9 9 9 9 9 9 9 9 9 9

What does the following program print?

```
#include <iostream>
using namespace std;
int main()
{
  int i;
  // change the next line in #5, #6
  int my_array[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

  for (i = 0; i < 10; i++)
    cout << my_array[i] << "  "; // change this line in #7
  return 0;
}
```

answer: 0 1 2 3 4 5 6 7 8 9

5. Change the declaration to

```
int my_array[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
```

What happens?

answer: error too many int values

6. Change the declaration to

```
int my_array[10] = { 0, 1, 2, 3, 4 };
```

What happens?

answer: 0 1 2 3 4 0 0 0 0 0

7. Change the `cout << my_array[i];` to `cout << my_array;`
What happens?

answer: 0OEBFEC8 0OEBFEC8 0OEBFEC8 0OEBFEC8 0OEBFEC8 0OEBFEC8 0OEBFEC8 0OEBFEC8 0OEBFEC8 0OEBFEC8

What does the following program print?

```
#include <iostream>
using namespace std;
int main()
{
  int i;
  int k = 55;
  int my_array[10];

  for (i = 0; i < 10; i++)
    my_array[i] = i;
  for (i = 0; i <= 12; i++)
    cout << my_array[i] << "  ";
  cout << endl;
  cout << " k = " << k << endl;
  return 0;
}
```

answer: 0 1 2 3 4 5 6 7 8 9 -858993460 -858993460 55
k = 55

9. What does the following program print? Where do the last two values come from?

```
#include <iostream>
using namespace std;
int main()
{
  int i;
  int my_array[10];
  int your_array[10];

  for (i = 0; i < 10; i++)
    your_array[i] = i * 20;
  for (i = 0; i < 10; i++)
    my_array[i] = i;
  for (i = 0; i <= 13; i++)
    cout << your_array[i] << "  ";
  return 0;
}
```

answer: 0 20 40 60 80 100 120 140 160 180 -858993460
-858993460 0 1

What happens when you compile the following program? How do you fix the problem so that both arrays contain the same content?

```
#include <iostream>
using namespace std;
int main()
{
  int i;
  int my_array[10];
  int your_array[10];

  for (i = 0; i < 10; i++)
    my_array[i] = i;
  your_array[:] = my_array[:];
  for (i = 0; i < 10; i++)
    cout << my_array[i] << "  ";
  cout << endl;
  for (i = 0; i <= 10; i++)
    cout << your_array[i] << "  ";
  return 0;
}
```

answer: error

0 1 2 3 4 5 6 7 8 9 -858993460 -858993460 -858993460 -858993460 -858993460 -858993460 -858993460 -858993460 -858993460 -858993460 -858993460

10. What does the following print?

```
#include <iostream>
using namespace std;

void add_one(int temp_array[5]);

int main()
{
  int i;
  int my_array[5] = { 1, 2, 3, 4, 5 };

  add_one(my_array);
  for (i = 0; i < 5; i++)
    cout << my_array[i] << "  ";
  return 0;
}

void add_one(int temp_array[ ])
{
  int j;

  for (j = 0; j< 5; j++)
    temp_array[j]++;
}
```

answer: 2 3 4 5 6

Is the array passed by reference or by value?

answer: pass by reference

In the function header and the function prototype, place an & in front of `temp_array[5]` What happens?

answer: array type of reference is not allowed

11. The following program passes elements into the function. What does the following print?

```
#include <iostream>
using namespace std;
void add_one(int temp_element);

int main()
{
  int i;
  int my_array[5] = { 1, 2, 3, 4, 5 };

  add_one(my_array[1]);
  for (i = 0; i < 5; i++)
    cout << my_array[i] << "  ";
  return 0;
}

void add_one(int temp_element)
{
  int j;
  for (j = 0; j < 5; j++)
    temp_element++;
}
```

answer: 1 2 3 4 5

Is this pass by reference or pass by value?

answer: by reference

What happens if you place an & in front of temp_element in the prototype and the function header?

answer: 1 7 3 4 5

Is this pass by reference or pass by value?

answer: by value

What does the program print if you leave off the subscript in the function call?

answer: _____

12. Write a function which fills an array, returning the number of elements entered, and a function that returns the **index** of the largest element in an array. The parameters for the filling function are the array and the number of elements in the array, for the largest index function are the array and the number of valid elements. **Do Not Modify** the **`main( )`** function except for filling in the parameters to the function calls.

```
#include <iostream>
using namespace std;

int fill_array(_____________________________________________);
int find_largest(____________________________________________);

int main()
{
  int   j;
  int   index;
  int   my_array[10];
  int   num_elements = 0;

  num_elements = fill_array(_____________________);

  index = find_largest(_____________________);
  cout << "The largest is:  " << my_array[index] << endl;

  return 0;
}

//fills array till full or user enters ctrl-z, whichever comes
//first
int fill_array(_____________________________________________)
{









}

//see next page for find_largest( )
int find_largest(_____________________________________________)
     {
```

```
}  // end of function find_largest
```