Question 1

1 / 1 point

What is the output of the following code?

```
int k = 6, m = 4;
int * p1 = &k;
int * p2 = &m;
p2 = p1;
p1 = &m;
*p2 = 7;
cout << k << " " << m << " " << *p1 << " " << *p2;
```

- 6 4 4 7
- 4 7 7 4
- 6 4 4 6
- ✓ 7 4 4 7

Question 2

1 / 1 point

Which statement is true about the code?

```
int k = 5;
int * ptr = &k;
```

- Variable **ptr** is constant and may not be reassigned
- Variable **ptr** holds the special null value
- ✓ Variable **ptr** holds the address of where variable **k** is stored
- Variable **ptr** holds the value 5

Question 3

1 / 1 point

What is the output of the code?

```
int x = 3;
```

```
int y = 4;
int * p1 = &x;
int * p2 = p1;
x = 5;
p1 = &y;
cout << x << " " << *p1 << " " << *p2;
```

- 3 4 5
- 5 5 5
- 3 4 3
- ✔ 5 4 5

Question 4 **1 / 1 point**

Which statement could be used to initialize a pointer to the null address?

- int * ptr = 0;
- int * ptr = NULL;
- int * ptr = nullptr;
- ✔ All choices are valid (depending on version of C++)

Question 5 **1 / 1 point**

Which line is invalid?

```
int k = 5;
const int * p1 = &k;
int * p2;
```

- p2 = &k;

- p1 = p2;
- k = 20;
- ✓ *p1 = 15;

Question 6 **1 / 1 point**

Which of the code lines is invalid?

- ✓
```
int k = 2;
int * p1 = &k;
double * p2 = p1;
```
-
```
int k = 2;
double d = 3.5;
int * p1 = &k;
double * p2 = &d;
*p2 = 0;
```
-
```
int k = 2;
double d = 3.5;
int * p1 = &k;
double * p2 = &d;
int m = (int)(*p2);
```
-
```
int k = 2;
double d = 3.5;
int * p1 = &k;
double * p2 = &d;
```

Question 7 **1 / 1 point**

Which line is invalid?

```
int k = 5;
int * const p1 = &k;
int * p2;
```

- *p1 = 15;
- p2 = &k;
- ✓ p1 = p2;

k = 20;

Question 8 **1 / 1 point**

In C++, a reference can be described as a constant pointer that is automatically dereferenced.

✓ True

False

Question 9 **1 / 1 point**

Given that **a** is an integer array starting at location 2000, **ptr** is a pointer to **a** and each integer is stored in 4 bytes of memory, what address is computed from: **(ptr + 4)**?

2000

2004

2012

✓ 2016

Question 10 **1 / 1 point**

Which statement best describes the array constructed from the initialization list?

int sales [12] = { 100 };

Array **sales** stores 100 in the first index with remaining elements undefined.

✓ Array **sales** stores 100 in the first index with remaining elements set to 0.

- Array **sales** stores 100 in each of the twelve indexes.
- Array **sales** is resized to one element and this one element stores 100.

Question 11 **1 / 1 point**

What is the output of the code?

```
int a [4];
a[0] = 7;
a[1] = 9;
a[2] = a[0] + a[1];
a[3] = a[2] – 1;
for (int k = 0; k < 4; k++)
  cout << a[k] << " ";
```

- 7 9 16 0
- ✔ 7 9 16 15
- 16 15 9 9
- 15 16 9 7

Question 12 **1 / 1 point**

A program compares respective array elements of two arrays. After the loop, it reports the number of elements that are equal to each other. Which code accomplishes this task? The code sample below would report count 2 since the first two elements are the same in each array.

```
int a[3] = { 10, 40, 30 };
int b[3] = { 10, 40, 50 };
int count = 0;
```

-
```
for (int k=0; k<3; k++)
        if (a == b) count++;
cout << count;
```

✔ for (int k=0; k<3; k++)
  if (a[k] == b[k]) count++;
cout << count;

for (int k=0; k<3; k++)
{ if (a[0] == b[0]) count++;
 if (a[1] == b[1]) count++;
 if (a[2] == b[3]) count++;
}
cout << count;

for (int k=0; k<3; k++)
{ if (a[k] == b[k]) count++;
 cout << count;
}

Question 13 **1 / 1 point**

Given an integer is stored in 4 bytes, the array declared below would use _____ bytes.

int values [5];

4

5

16

✔ 20

Question 14 **1 / 1 point**

Array **list2** should contain the same elements as array **list1**. The code below correctly accomplishes this task.

```
int list1 [ ]  = {10, 20, 30, 40};
int list2 [4];
list2 = list1;
```

True

✓ ○ False

Question 15 **0 / 1 point**

Which statement is true about inserting 22 into the partially filled array shown using the most efficient algorithm?

20	10	60	5	70		

➡ ○ 22 should be inserted at index 5

○ The current elements in indexes 0..4 should be moved one to the right and 22 should be inserted at index 0

○ 22 should be inserted at index 6

✗ ○ 22 should be inserted at index 0, overwriting value 20

Question 16 **1 / 1 point**

Which function prototype(s) could be used for a function that is passed the base address of an int array and its capacity?

○ void display (int a [ ], int capacity);

○ void display (int * a, int capacity);

✓ ○ Both function prototypes are acceptable

○ Neither function prototype is acceptable

Question 17 **1 / 1 point**

Which statement is not true about the function prototype?
void mystery (int a [ ], int size);

- Parameter **size** is a value parameter
- The function will be able to alter the argument array associated with parameter **a**
- ✓ This function is called from **main** by passing in the first element of argument array **b** with capacity 10 as: **mystery (b[0], 10);**
- Parameter **a** receives the base address of the argument array

Question 18 **0 / 1 point**

Which function prototype is correct for a function that is passed an integer array, its size (number of elements used) and an integer target, such that, when the function completes, all occurrences of the target in the argument array have been removed?

- ✗ void remove(int a[ ], int size, int target);
- void remove(const int a[ ], int & size, int target);
- void remove(const int a[ ], int size, int target);
- ➡ void remove(int a[ ], int & size, int target);

Question 19 **1 / 1 point**

During run-time of its function, the array declared below will have memory storage allocated for 10 integers.

int a [10];

- ✓ True
- False

Question 20 **1 / 1 point**

What is the output of the code?

int a[ ] = { 10, 20, 30 , 40, 50, 60 };

```
int* ptr = a + 1;
ptr++;
ptr = ptr + 2;
cout << *ptr;
```

- 30
- 40
- ✔ 50
- A syntax error occurs