

## Task 1

- 1.1) a: Stack, memory automatically allocated at compile time.  
b: Heap, explicit memory allocation using the "new" keyword.  
c: Stack, fixed sized memory allocation.  
n: Stack, memory automatically allocated at compile time.  
d: Heap, memory allocated at runtime.  
e: Stack, a constant memory address has been allocated, and does not point to any data.  
f: Stack, a pointer to a memory address has been allocated.  
g: Stack, memory automatically allocated at compile time.  
h: Stack, memory automatically allocated at compile time.  
c[10]: Stack, it is a pointer to the variable "e", memory allocated at compile time.
- 1.2) Assigning NULL to an constant integer will compile, however it will not be very useful since the value cannot change.
- 1.3) "int a;" -> Without initializing a, it will contain a garbage value. If it is then used for calculations, it may yield unexpected results.
- "char g = 2;" -> This will assign the character corresponding to the ASCII value of 2 rather than 2 itself. It may be best to change it to "char g = '2';"
- "c [ 10 ] = \*&\*e;" -> The array "c" was assigned a size of 10. This is accessing a garbage value, since it is out of bounds.

## Task 2

- 2.1) First (before the derived class's constructor).
- 2.2) Last (after the derived class's destructor).
- 2.3) After.
- 2.4) Class A, then Class B.
- 2.5) Class B, then Class A.

## Task 3

- 3.2) It worked, these are numeric types that support the "/" operator. It performed integer division.
- 3.3) It worked, however the output was truncated. Assigning floats to integers truncates the values.
- 3.4) It did not work, since n1 and n2 are integers, the values "Hello", "World", and "!" cannot be stored in them.
- 3.5) It did not work, same reason as above.

## Task 4

4.1 a) 15 15

4.1 b) 15 4

4.1 c) 15 15

4.1 d) 15 15

4.1 e) [random memory address] 15

## Task 6

6.1) AuditableSnapshot

6.2) Snapshot

6.3) User

6.4) UserManager

6.5) See code

6.6) See code