**Concepts covered**

* **Variable types**
* **Logic**
* **Loops**
* **Integer division**
* **Files**
* **Vectors & Arrays**
* **Memory & Pointers**

**Other Notes**

* I sometimes use non-standard / goofy bracket placement to save space
* JavaScript questions often use alert() -> treat like C++ cout

**2. [1] Which of the following \*is\* a valid variable name?**

1. ninety\_nine\_balloons
2. ninety nine balloons
3. 99\_balloons
4. None of the above
5. All of the above

**4. [1] Name the underlined section in the following function definition:**

|  |
| --- |
| double foo(int bar) |

1. Prototype
2. Function
3. Parameter
4. Return Value

**5. [1] Which of the following \*IS NOT\* a relational operator (used in IF statements)**

1. <
2. ==
3. !=
4. =
5. >=

**. [2] How many times will this loop?**

|  |
| --- |
| int i = 0;  while(i < 5)  {  i += 3  } |

1. 0
2. 1
3. 2
4. 3
5. 4

None of the above

**8. [2] What is the value of x after this loop?**

|  |
| --- |
| int x = 2;  int i = 0;  for(i = 0; i < 4; i++)  {  x = x \* x;  } |

1. 0
2. 8
3. 16
4. 64
5. 256
6. 512
7. None of the above

**14. [1] What is the value of "x"? BE SURE TO LOOK CAREFULLY AT THE FLOW OF EXECUTION!!**

|  |
| --- |
| int modify(int x)  {  return 0;  x = 5;  return x;  }  int main(void)  {  int x = 3;  x = modify(x);    //x now equals what?  } |

1. 0
2. 2
3. 3
4. 5
5. None of the above

**17. [2] How many times will "Hello" be printed to the screen?**

|  |
| --- |
| **int i = 0;**  **for(i = 3; i < 10; i = i \* i)**  **{**  **printf("Hello\n");**  **}** |

1. 0
2. 1
3. 2
4. 3
5. 9
6. None of the above

**19. [2] What is the value of x after the FOR loop completes?**

|  |
| --- |
| **int x = 1;**  **int i = 0;**  **for(i = 1; i < 4; i++)**  **{**  **if(i \* x < 5)**  **{**  **x++;**  **}**  **else if(i / 2 == 0)**  **{**  **x = x \* x;**  **}**  **else**  **{**  **x += i;**  **} }** |

**21. [2] What does the following code segment do? In other words, what happens to numbers[] when the loop finishes?**

|  |
| --- |
| **int numbers[] = {1, 2, 3, 4, 5, 6};**  **int temp = 0;**  **int i = 0;**  **for(i = 0; i < 3; i++)**  **{**  **temp = numbers[i];**  **numbers[i] = numbers[5 - i];**  **numbers[5 - i] = temp;**  **}** |

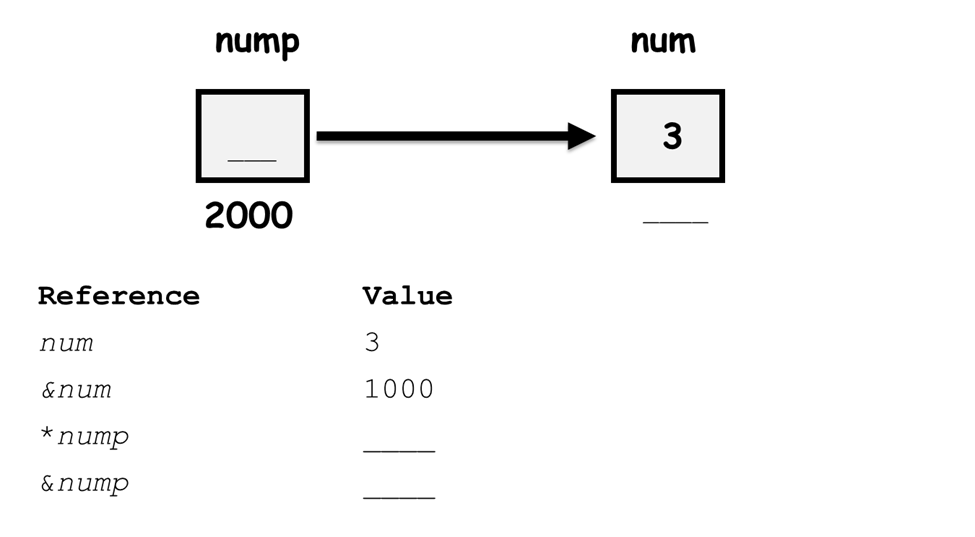
**22. [2] Assume that we declare the following variables:**

|  |
| --- |
| **double weight = 150.3;**  **int ages[3] = {0};**  **int\* ptr = &ages[1];** |

**Use the cubbyhole diagram metaphor to diagram how these variables will be represented in memory:**

**23. [1] What is the relationship between & and \* when dealing with pointers?**

**25. [2] Fill in the missing parts of the diagram below. Missing parts are indicated by underscores (\_).**



**26. [2] Identify (circle and explain) incorrect portions of the following code segment.**

//will turn a lowercase a into a capital A

void capitalize\_a(char \*some\_char)

{

if(some\_char = "a")

{

//lowercase a is exactly 32 below upper case A

some\_char += 32;

}

return some\_char;

}

**27. [2] What is the value of "foo" inside of main?**

|  |
| --- |
| void swap(int \*left, int \*right)  {  int temp = \*left;  \*left = \*right;  \*right = temp;  }  int main(void)  {  int foo = 5;  int other = 2;  int \*foo\_ptr = &foo;  swap(foo\_ptr, &other);  return 0;  } |

**29. [5] Consider the following chunk of code:**

|  |  |
| --- | --- |
| if(a < b && c > a)  {  if(b < c)  {  printf("Spam Please!\n");  }  else  {  printf("It's a late parrot!\n");  }  }  else if(b > c && b > a)  {  printf("Cheese Shoppe\n");  if(a > c)  {  printf("Cheddar\n");  }  else if(a < c)  {  printf("Gouda\n");  }  else if(a == c)  {  printf("Swiss\n");  }  }  (code continues in next column) | else  {  printf("Trees\n");  if(a == b)  {  printf("Chestnut\n");  }  else  {  printf("Larch\n");  }  }  printf("Done"); |

**Complete the following table given the values of a, b, and c**

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | Result |
| 1 | 1 | 0 |  |
| 1 | 2 | 3 |  |
| 1 | 2 | 1 |  |
| 1 | 0 | 1 |  |
| 1 | 5 | 3 |  |