// Dion Niazi dn3gy 14 02 2017 inlab4.pdf
What is your name?
Dion Niazi
What is your quest?
To finish this worksheet and get an A for the class
What is your favorite color?
Green

Size of C++ data types

C++ Type	Size in bytes?	Max value? (base 10)	Zero is stored as (in hex)?	One (or 1.0) is stored as (in hex)?
int	4	2147483647	0x00000000	0x00000001
unsigned int	4	4294967295	0x00000000	0x00000001
float	4	3.40282e+38	0x00000000	0x0000038F
double	8	1.79769e+308	0x0000000000000000	0x3FF0000000000000
char	1	255	Char'0'= 0x30	Char '1'= 0x31
bool	1	1	false = 0x00	true = 0x01
C++ Type	Size in bytes?	Max value? (base 16 (hexadecimal))	NULL is stored as?	
int*	8	0xFFFFFFFFFFFFFFFF	0x0000000000000000	
char*	8	0xFFFFFFFFFFFFFFFF	0x0000000000000000	
double*	8	0xFFFFFFFFFFFFFFFF	0x0000000000000000	

Primitive Arrays in C++

How does the compiler determine the address of **`&(IntArray2D[i][j])`**? Assume the array is defined as: **`int IntArray2D[6][5];`**
First the complier gets the base address of the array, which would be the first element of the array (IntArray[0][0]). Then the complier will add the product of the size of the elements, so in this case the size would be 4 bytes because it's an integer, to the base address and the sum of the element j and the product of i and the size of j elements, which

is 5 in this case. But when you add to the base address you first need to convert the base 10 number into base 16 and then add to get the memory location of IntArray2D [6][5].

Memory of IntArray2D [6][5] = base Address + byte size * (i*j.size() + j)

Memory of IntArray2D [6][5] = base Address + 4 * (i*5 + j)