Basics3 – Pointers

Due Date

- See Piazza for due date and time
  - Grading the next day
- Submit program to perforce in your student directory
  - Sub directory called:
    - /Basics3/...
  - Fill out your ***Basics3 Submission Report.pdf***
    - Place it in the same directory as your solution
    - Enter the final Changelist number of your submission
    - Enter the number of test passed
    - Write up a quick discussion in the report
      - What you learned from this basics

Goals

- C++ pointers
  - Saving the world one dereference at a time.
  - Increasing C++ knowledge and understanding

Assignments

- General:
  - Add code to the body of the functions:
    - Students_PointerWalk()
    - Students_Casting()
  - Run the Unit Tests to verify progress / success
    - 5/5 is the best for this program

- Students_PointerWalk()
  - Code up the pointer test from class (See Below)
    - Please code and step through each of these steps
    - Verify with break points and memory windows
    - This is for your benefit.
    - Please do so...

- Students_Casting()
  - Understand the 3 structures, Cat, Bird, and Dog.
  - Understand how they are added arranged inside the petStore structure.
    - Pay particular attention to the padding and alignment
  - Code the questions 1-19
    - Restrict your answers to the rules/guidelines presented in code

  - You should be able to answer those questions by paper first
    - Then verify with the code.
    - Make sure you understand these questions / relationships.

- Make sure that your program compiles and runs
  - Warning level 4 sometimes that is not possible due to MS headers…
  - Your code should be squeaky clean.

- Submit program to perforce in your student directory
  - Sub directory called: /Basics3/…

Validation

Simple check list to make sure that everything is checked in correctly

- Did you do all run all unit tests problems?
- Do they compile and run without any errors?
- Warning level 4 free?
- Submitted it into /Basics3 directory - without the extra files?
- Submit the submission report?
- Can you delete you local drive, regrab the Basics3 directory?
  - Is all the code there?

Hints

Most assignments will have hints in a section like this.

- This is pretty easy Basic assignment
  - It is mainly here to help you single step through your code and understand pointers layouts and access commands.
  - The casting section, allows you to access parts of an complicated structure with casting.
    - Note the data is the same, but the way you access changes.
- I expect this assignment to be completed quickly for most of the students
  - Please make sure you fully understand this code without a debugger.
  - Many little lessons here for those who put in the effort.
- Enjoy

Pointer Test / Keenan

```
Assume that we are working on a LITTLE endian processor
unsigned char data[];
Memory Dump ( values in Hex )

data =  0x0000:  AB CD 12 3F
        0x0004:  33 B5 D3 35
        0x0008:  23 24 01 FE
        0x000C:  CD 33 44 55
        0x0010:  66 03 75 33
        0x0014:  29 55 22 11
        0x0018:  56 88 A9 13
        0x001C:  14 82 68 26

unsigned char  *p;  // char are 8-bits wide
unsigned int   *r;  // ints are 32-bits wide
unsigned short *s;  // shorts are 16-bits wide

p = &data[0];                          Expected output
printf("%x\n",  *(p+3)  );     1)_____________________________
printf("%x\n",  *(p+5)  );     2)_____________________________
p = p + 12;
printf("%x\n",   *(p)  );      3)_____________________________
printf("%x\n",   p[2]  );      4)_____________________________
printf("%x\n",   *p++  );      5)_____________________________
p += 6;
printf("%x\n",   *--p  );      6)_____________________________
printf("%x\n",   p[5]  );      7)_____________________________
p = p + 2;
printf("%x\n",   *p++  );      8)_____________________________
printf("%x\n",  *(p+3) );      9)_____________________________
p = 5 + p;
printf("%x\n",  *(p++) );     10)_____________________________
printf("%x\n",  *(--p) );     11)_____________________________
```

```
data =  0x0000:  AB CD 12 3F
        0x0004:  33 B5 D3 35
        0x0008:  23 24 01 FE
        0x000C:  CD 33 44 55
        0x0010:  66 03 75 33
        0x0014:  29 55 22 11
        0x0018:  56 88 A9 13
        0x001C:  14 82 68 26

r = (unsigned int *)&data[0]
printf("%x\n",   *(r)  );        12)_____
printf("%x\n",  *(r+5) );        13)_____
r++;
printf("%x\n",   *r++  );        14)_____
r = r + 2;
printf("%x\n",   r[2]  );        15)_____
r = r + 1;
printf("%x\n",   r[0]  );        16)_____

s = (unsigned short *) r;
printf("%x\n",   s[-2] );        17)_____
s = s - 3;
printf("%x\n",   s[2]  );        18)_____
s += 5;
printf("%x\n",  *(s+3) );        19)_____
printf("%x\n",   *(s)  );        20)_____
p = (unsigned char *) s;
printf("%x\n",  *(p+3) );        21)_____
p += 5;
printf("%x\n",  p[-9] );         22)_____
--p;
printf("%x\n",   p[0] );         23)_____
```