

# Traditional Pointer Solutions

- Write a simple program that creates a pointer
- Use a stack variable to set the pointer's value
- Print out the value of the pointer
- Print out the value of the stack variable by dereferencing the pointer

- Explain what is meant by passing an argument by address to a function
  - Instead of making a copy of a variable (pass by value), the program uses its address
  - This means that any changes made to the argument in the function body will also be made to the variable in the caller

- Write a simple program which passes an argument by address to a function which modifies it
- In the function, print out the value of the pointer and the data it points to, before and after modifying it
- Print out the value of the variable in main() before and after calling the function

- Explain how to obtain allocated memory from the heap
  - Use the new operator
- Why are pointers needed for this kind of memory?
  - new returns the address of the allocated memory, which can only be stored in a pointer
- What important thing must you do with this memory when you have finished using it? Why is it important?
  - Memory should be released by the delete operator
  - To avoid a “memory leak”

- Explain how to obtain allocated memory from the heap which can be used for an array
  - Use the array form of new
- Write a simple program which allocates memory from the heap for an array. Populate the array and print out the values of its elements
- What important thing must you do with this memory when you have finished using it? Why is it important?
  - Use the delete[] operator to prevent a memory leak