C++ Warmup on Parameter Passing / Peer Response Writing Techniques

CSCI-3081: Program Design and Development

Parameter Passing Example

- Will this work?

```
// Exchange the values of v1 and v2
void swap(int v1, int v2) {
   int tmp = v1;
   v1 = v2;
   v2 = tmp;
}
```

- By default C++ uses **pass-by-value**, which means the called routine cannot modify the original ("outside") value of v1 and v2.

Example

- We can make it work with pointers because now we're modifying what v1 and v2 point to rather than the actual values of v1 and v2.

```
// Exchange the values of v1 and v2
void swap(int *v1, int *v2) {
  int tmp = *v1;
  *v1 = *v2;
  *v2 = tmp;
}
```

- But, there's a better way in C++.

Parameter passing

- In C++, all parameters are passed by value unless you specify otherwise.
- With **pass-by-value**, the called routine cannot modify the original value.
- With **pass-by-reference** (indicated by "&") you can.

```
// Exchange the values of v1 and v2
void swap(int &v1, int &v2) {
  int tmp = v1;
  v1 = v2;
  v2 = tmp;
}
```

Three Versions of Swap

Pass by Value

(Doesn't Work)

```
void swap(int v1, int v2) {
  // swaps v1 and v2
  …
}

int main() {
  int a = 1;
  int b = 2;
  swap(a,b);

  // do something with a and b
  // assuming they have been swapped
}
```

Three Versions of Swap

Pass Using Pointers

(Works, but a bit messy)

```
void swap(int *v1, int *v2) {
  // swaps v1 and v2
  …
}

int main() {
  int a = 1;
  int b = 2;
  swap(&a,&b);

// Or, you could also do:

  int *a = NULL;
  int *b = NULL;
  a = new int;
  b = new int;
  *a = 1;
  *b = 2;
  swap(a,b);
  delete a;
  delete b;
}
```

Three Versions of Swap

Pass by Reference

(Best)

```
void swap(int &v1, int &v2) {
  // swaps v1 and v2
  …
}

int main() {
  int a = 1;
  int b = 2;
  swap(a,b);
}
```

Pass-by-Reference

- What's good about references?
  - Our code is a bit cleaner than it would be if we used pointers -- we don't need the extra syntax of * and ->
  - Unlike pointers which can be NULL, there's no such thing as a null reference.

Pass-by-value with pointers

- Using pointers you can modify the “outside” data, even when using pass-by-value -- what’s really happening?
- If you pass a pointer by value, it still passes a value, but it’s the pointer value, i.e. the location in memory.
- So, we can’t change the value, only what it points to.

```
// Exchange the values of v1 and v2
void swap(int *v1, int *v2) {
  int tmp = *v1;
  *v1 = *v2;
  *v2 = tmp;
}
```

Diagramming Function Calls

- We can use diagramming to help understand pass by reference and value and the impact on memory.
- Example:

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}

int x = 2;
int y = foo(&x, 4+3);
```

Diagramming Function Calls

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}
...
int x = 2;
int y = foo(&x, 4 + 3);
```

Step 1: Allocate space for x and y.

x	
y	

Diagramming Function Calls

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}
...
int x = 2;
int y = foo(&x, 4 + 3);
```

Step 2: Initialize x.

x	2
y	

Diagramming Function Calls

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}
...
int x = 2;
int y = foo(&x, 4 + 3);
```

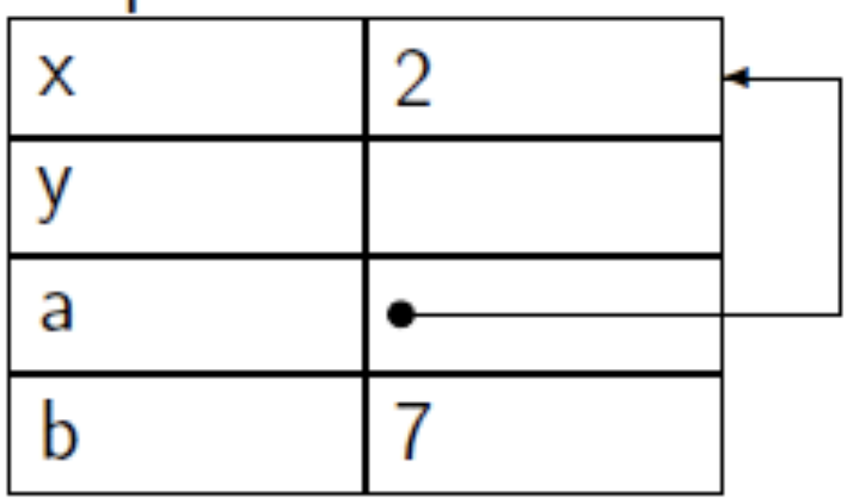
Step 3: Call `foo`, allocate space for `a` and `b`.

x	2
y	
a	
b	

Diagramming Function Calls

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}
...
int x = 2;
int y = foo(&x, 4 + 3);
```

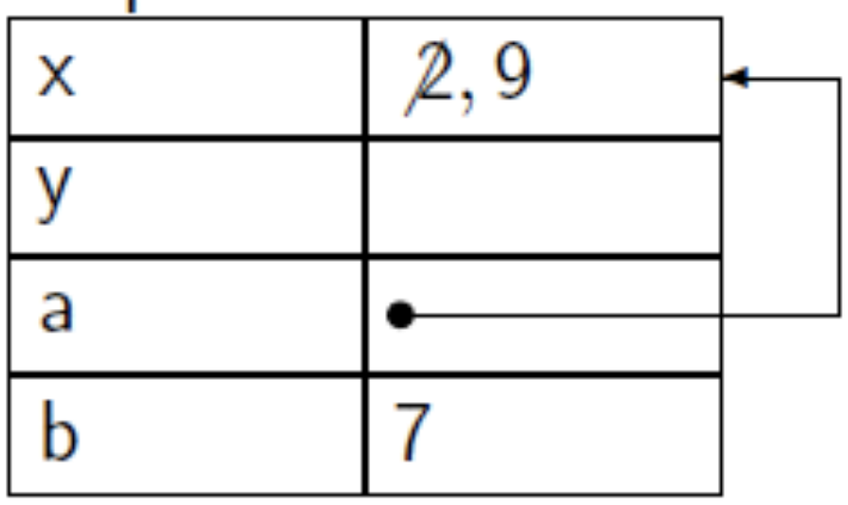
Step 4: Initialize a and b.



Diagramming Function Calls

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}
...
int x = 2;
int y = foo(&x, 4 + 3);
```

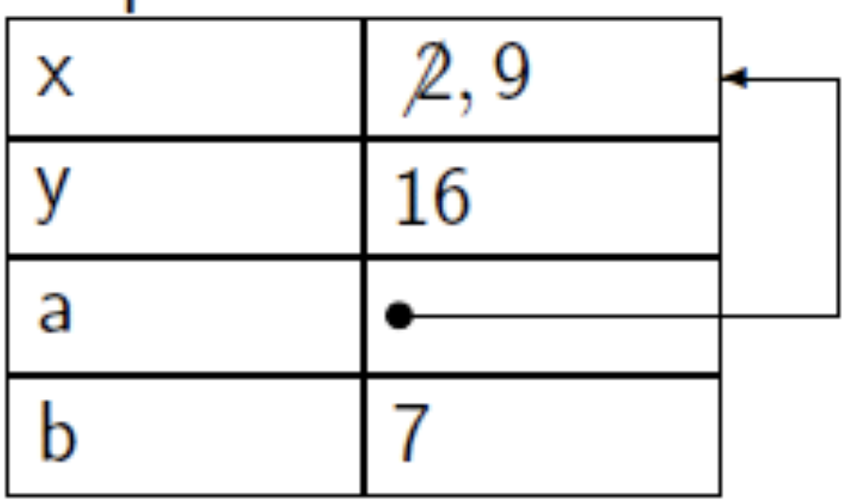
Step 5: Execute `*a = *a + b;`



Diagramming Function Calls

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}
...
int x = 2;
int y = foo(&x, 4 + 3);
```

Step 6: Execute `return *a + b;`



Diagramming Function Calls

```
int foo(int *a, int b) {
  *a = *a + b;
  return *a + b;
}
...
int x = 2;
int y = foo(&x, 4 + 3);
```

Step 7: Clean up after `foo` – deallocate `a` and `b`.

x	~~2~~, 9
y	16

Exercise

```
int bar(int x, int *y) {
  int *z;
  z = &x;
  *z = *y * *z;
  return *z;
}

…

int a = 6;
int b = bar(a, &a);
```

Peer Response Writing Techniques

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

What I want you to learn:

- A process to help you usefully respond to writing that describes, documents, or otherwise has something to do with software development.
- How to use what you learn to improve your own writing about software development.
  - Mimic the good things that you see.
  - Avoid the bad things that you see.
  - Critique your own writing following the same process.

Let's start by picking some example "peer" writing so that we have something to respond to:

Google C++ Style Guide

Why does Google need a C++ style guide?

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Let's find out:

http://google-styleguide.googlecode.com/svn/trunk/cppguide.html

Your Activity - Peer Response

Give us your own response to this writing by marking up the paper version that we're handing out. Here are four specific prompts:

Skimability:

- When you opened this document, where did your eyes go first (star this section)?
- What questions arose? (Jot them down beside the section that prompted them.)
- Where did you stop skimming and start reading? Why?

Audience:

- What sort(s) of readers is this document geared to? How can you tell?
- Underline any portions of the document in which the text jumps to a different target reader (i.e., where understanding the content might require a different background).

Purpose:

- Choose two sections and list questions that were answered by the text.
- What un-addressed questions occurred to you as you were reading each section?

Finally:

- What is the most successful element of this document?
- What are the 2-3 most salient revision suggestions you have for its author?

Your Activity - Peer Response

Give us your own response to this writing by marking up the paper version that we're handing out. Here are four specific prompts:

Skimability:

- When you opened this document, where did your eyes go first (star this section)?
- What questions arose? (Jot them down beside the section that prompted them.)
- Where did you stop skimming and start reading? Why?

Audience:

- What sort(s) of readers is this document geared to? How can you tell?
- Underline any portions of the document in which the text jumps to a different target reader (i.e., where understanding the content might require a different background).

Purpose:

- Choose two sections and list questions that were answered by the text.
- What un-addressed questions occurred to you as you were reading each section?

Finally:

- What is the most successful element of this document?
- What are the 2-3 most salient revision suggestions you have for its author?

Your Activity - Peer Response

Give us your own response to this writing by marking up the paper version that we're handing out. Here are four specific prompts:

Skimability:

- When you opened this document, where did your eyes go first (star this section)?
- What questions arose? (Jot them down beside the section that prompted them.)
- Where did you stop skimming and start reading? Why?

Audience:

- What sort(s) of readers is this document geared to? How can you tell?
- Underline any portions of the document in which the text jumps to a different target reader (i.e., where understanding the content might require a different background).

Purpose:

- Choose two sections and list questions that were answered by the text.
- What un-addressed questions occurred to you as you were reading each section?

Finally:

- What is the most successful element of this document?
- What are the 2-3 most salient revision suggestions you have for its author?

Your Activity - Peer Response

Give us your own response to this writing by marking up the paper version that we're handing out. Here are four specific prompts:

Skimability:

- When you opened this document, where did your eyes go first (star this section)?
- What questions arose? (Jot them down beside the section that prompted them.)
- Where did you stop skimming and start reading? Why?

Audience:

- What sort(s) of readers is this document geared to? How can you tell?
- Underline any portions of the document in which the text jumps to a different target reader (i.e., where understanding the content might require a different background).

Purpose:

- Choose two sections and list questions that were answered by the text.
- What un-addressed questions occurred to you as you were reading each section?

Finally:

- What is the most successful element of this document?
- What are the 2-3 most salient revision suggestions you have for its author?

(Returning to…) What I want you to learn:

- A process to help you usefully respond to writing that describes, documents, or otherwise has something to do with software development.
- How to use what you learn to improve your own writing about software development.
  - Mimic the good things that you see.
  - Avoid the bad things that you see.
  - Respond to your own writing following the same process.

Reminders: What's happening this week?

Project:

- You completed your individual initial design for a project solution diagrammed via UML.
- Now, you should be working full force as a team to refine this design and implement the project.

Weekly writing:

- Peer response building upon today's exercise.

Recent class meetings:

- C++ technical detail (see Eckel's book for supplemental reading on pointers, polymorphism, parameter passing, etc.).
- Thursday we'll begin a section on Design Patterns - exciting!

Labs:

- This week is the first of the labs reserved for group work on the project.