BYU
IDAHO

Structs – Deep Dive

CS 165 – Object Oriented Software Development

Macbeth – Lesson 2.3

Agenda

- Opening Prayer
- Music Friday
- Q&A
- Nested Structs
- Passing Structs in Functions
- Dynamically Allocated Structs
- Looking Forward

BYU IDAHO

Music Friday

Nearer, My God, to Thee (Hymn 100)

Nearer, my God, to thee,
Nearer to thee!
E'en though it be a cross
That raiseth me.
Still all my song shall be
Nearer, my God, to thee,
Nearer, my God, to thee,
Nearer to thee!

Points and Lines

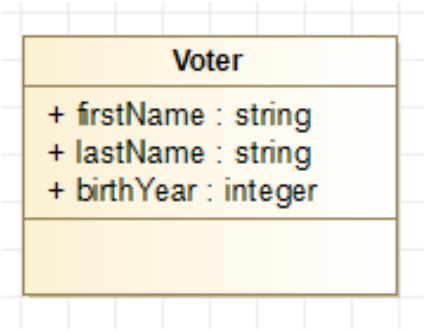
```
struct Point
{
   double x;
   double y;
};


struct Line
{
   Point p1;
   Point p2;
   int color;
};
```

Searching for Lost Sheep

Voter Registration Rolls (public records!)

```
struct Voter
{
   string firstName;
   string lastName;
   int birthYear;
}
```

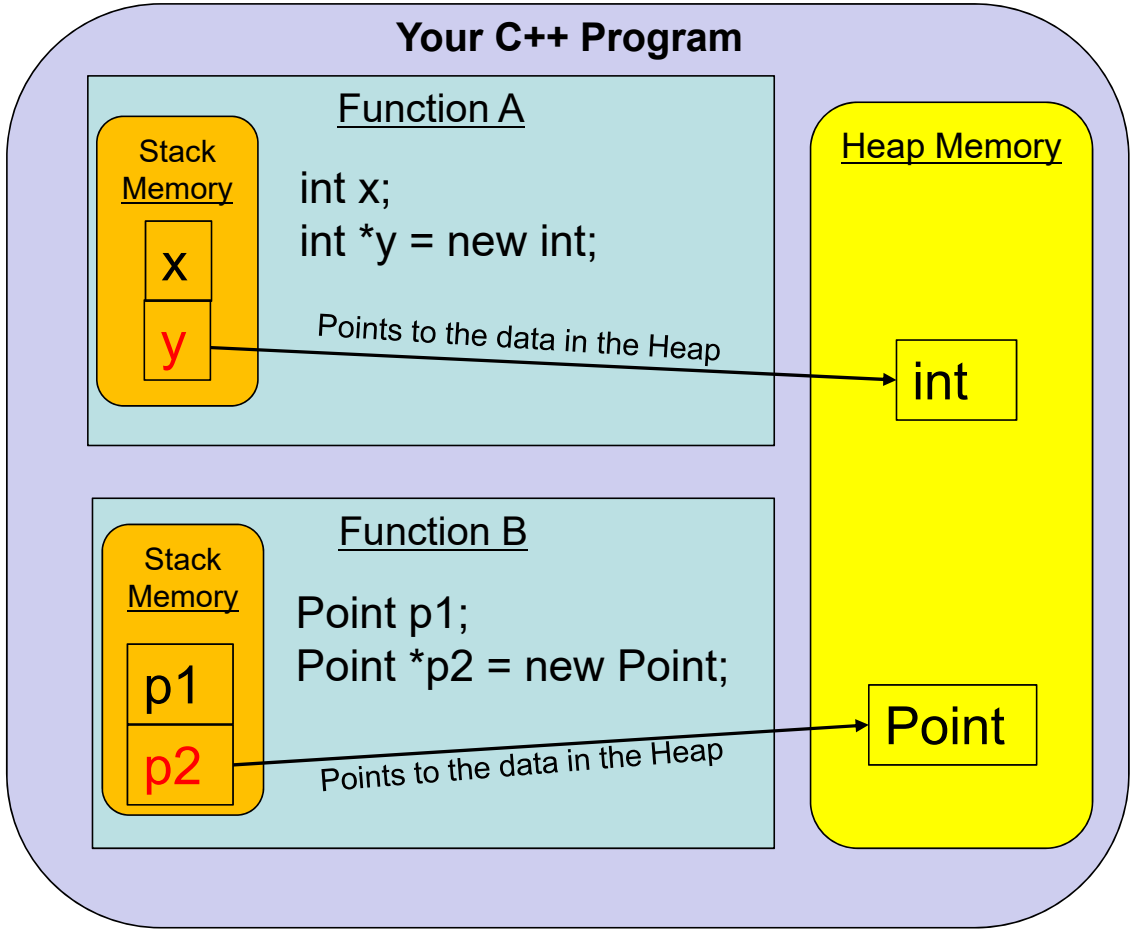


Write a function called `findVoterLastName`:

- Inputs: an array of Voter's, the size of the array, and the last name to match
- Outputs: an array of Voter's that match the last name and the size of the result array

Stack and Heap

- Every function has stack memory which stores all your variables.
- If you use the “new” keyword, then the variable is stored in the heap.
- The function stack memory needs to maintain a pointer to the variable in the heap (a.k.a don’t lose it!)



Allocating Structures

	Statically Allocated Structure	Dynamically Allocated Structure
Where is it stored?	Stack Memory in the Function	Heap Memory in the Program
When is it stored?	At Compile Time	At Run Time

Example
 	
`void disp(Point *p)`
`{`
`  if (p == NULL)`
`  {`
`  return;`
`  }`
`  cout << p->x`
`  << " "`
`  << p->y;`
`}`
	`Point p1;`
	
	`p1.x = 2;`
	`p1.y = 3;`
	
	`disp(&p1);`
		`Point *p2 = new Point;`
		
		`p2->x = 2;`
		`p2->y = 3;`
		
		`disp(p2);`
		
		`delete p2;`

Looking Forward

- Monday
  - 02 Prove – Assignment (Digital Forensics) by 11:59pm

BYU IDAHO