Structs and Graphs

Joel Willoughby

COP 3503

Outline

Agenda

- Talk about some of the recurring problems from last lab
- Structs
- Graphs

Outline

while(1) break;

```
while(1){
  // Useful Code
  if(conditional)
    break;
  // Other Useful Code
}
```

- I saw this a few times last lab

while(1) break;

```
while(1){
  // Useful Code
  if(conditional)
    break;
  // Other Useful Code
}
```

- I saw this a few times last lab
- Only use infinite loops if you need them to run forever!

while(1) break;

```
while(1){
  // Useful Code
  if(conditional)
    break;
  // Other Useful Code
}
```

- I saw this a few times last lab
- Only use infinite loops if you need them to run forever!
- Notice how in this case we know when the loop ends

while(1) break;

```
while(1){
  // Useful Code
  if(conditional)
    break;
  // Other Useful Code
}
```

- I saw this a few times last lab
- Only use infinite loops if you need them to run forever!
- Notice how in this case we know when the loop ends

```
while(conditional){
  // All the Useful Code
}
```

char y or ‘y’

- char y;
  A vairable of type char called y. It has no special meaning
- ‘y’
  The character ‘y’. The program treats this specially (like 1 is treated as an integer 1)

char y or ‘y’

- char y;
  A vairable of type char called y. It has no special meaning

- ‘y’
  The character ‘y’. The program treats this specially (like 1 is treated as an integer 1)

```
if(choice == 'y'){
  // Do stuff
}

if(choice == y){
  // Do stuff
}
```

== or =

- == is a comparison operator, like <, >, <=, >=, !=
- = is an assignment operator

== or =

- == is a comparison operator, like <, >, <=, >=, !=
- = is an assignment operator
- Typically, you would not use = in a conditional:

```
if(choice = 'y') //Usually not how you wanted
```

== or =

- == is a comparison operator, like <, >, <=, >=, !=
- = is an assignment operator
- Typically, you would not use = in a conditional:

```
if(choice = 'y') //Usually not how you wanted
```

```
if(choice == 'y') //Makes sense
```

Code formatting

- Formatting is something you should always do.
- Even if you are using a new editor, you should still format correctly
- Don't become handicapped to one editor's style

```
int main(){
while(cond){
//Do some code
if(cond2){
//Do some other code...
}
//More loop code
}
}
```

Outline

Goal

- We will implement a graph structure
- Also, implement a menu
- Hopefully, these things will help with your project 1
- No file I/O this lab

Structs syntax

```
struct Graph{
  string vertices[MAX_VERTICES];
  int edges[MAX_EDGES][2];
  int numEdges;
  int numVertices
};
```

- Structs are ways to put data together
- Graph is now a type name
- Can access elements with (.) for references and (->) for pointers

```
Graph graph;
Graph * graphPtr = &graph;
graph.numVertices = 0;
graphPtr->numEdges = 0;
```

Structs cont.

- structs are very much like classes
- They can have constructors and member functions
- Big difference is elements are by default public (as opposed to private)

Graphs

```
struct Graph{
  string vertices[MAX_VERTICES];
  int edges[MAX_EDGES][2];
  int numEdges;
  int numVertices
};
```

- An aside on graphs and this representation (done on the board)

Outline

Questions

???