Fundamentals of Computing: Lecture 33

Piyush P Kurur
Office no: 224
Dept. of Comp. Sci. and Engg.
IIT Kanpur

October 28, 2009

Summary of last class

Summary of last class

- C preprocession (cpp)

Summary of last class

- C preprocession (cpp)
- Including files

Summary of last class

- C preprocession (cpp)
- Including files
- Defining macros.

Including files

```
#include <stdio.h>
#include <openssl/x509.h>
#include "foo.h"
```

Including files

```
#include <stdio.h>
#include <openssl/x509.h>
#include "foo.h"
```

- The `#include` line is replaced by the contents of the file

Including files

```
#include <stdio.h>
#include <openssl/x509.h>
#include "foo.h"
```

- The `#include` line is replaced by the contents of the file
- The general syntax is

Including files

```
#include <stdio.h>
#include <openssl/x509.h>
#include "foo.h"
```

- The `#include` line is replaced by the contents of the file
- The general syntax is

```
#include <filepath>

#include "filepath"
```

Including files

```
#include <stdio.h>
#include <openssl/x509.h>
#include "foo.h"
```

- The `#include` line is replaced by the contents of the file
- The general syntax is

  ```
  #include <filepath>
  ```

  ```
  #include "filepath"
  ```

- When angle backets are used the the file path is relative to a standard directory usually `/usr/include/`.

Including files

```
#include <stdio.h>
#include <openssl/x509.h>
#include "foo.h"
```

- The `#include` line is replaced by the contents of the file
- The general syntax is

  ```
  #include <filepath>
  ```

  ```
  #include "filepath"
  ```

- When angle backets are used the the file path is relative to a standard directory usually `/usr/include/`.
- When double quotes the file path is relative ot the current director.

Macro definitions

Macro definitions

```
#define ANSWER 42
#define ANSWERSTR  "The answer is 42"
#define mul(a,b)  ((a) * (b))
```

Macro definitions

```
#define ANSWER 42
#define ANSWERSTR  "The answer is 42"
#define mul(a,b)  ((a) * (b))
```

Macro definitions

```
#define ANSWER 42
#define ANSWERSTR  "The answer is 42"
#define mul(a,b)  ((a) * (b))
```

- The macros are substituted literally.

Macro definitions

```
#define ANSWER 42
#define ANSWERSTR  "The answer is 42"
#define mul(a,b)  ((a) * (b))
```

- The macros are substituted literally.
- Happens before compilation

Macro definitions

```
#define ANSWER 42
#define ANSWERSTR  "The answer is 42"
#define mul(a,b)  ((a) * (b))
```

- The macros are substituted literally.
- Happens before compilation
- Macros can take arguments

Macro definitions

```
#define ANSWER 42
#define ANSWERSTR  "The answer is 42"
#define mul(a,b)  ((a) * (b))
```

- The macros are substituted literally.
- Happens before compilation
- Macros can take arguments
- Macro with arguments should be used with care.

The static keyword

Syntax of usage

static declaration

```
static int x;
int foo()
{
  static int foo_invoke=0;
  foo_invoke ++;
  return foo_invoke;
}
```

The static keyword

Syntax of usage

static declaration

```
static int x;
int foo()
{
  static int foo_invoke=0;
  foo_invoke ++;
  return foo_invoke;
}
```

- A variable can be declared static.

The static keyword

Syntax of usage

static declaration

```
static int x;
int foo()
{
  static int foo_invoke=0;
  foo_invoke ++;
  return foo_invoke;
}
```

- A variable can be declared static.
- If the variable is a declared inside a function then all the invocation of the function uses the same variable.

The static keyword

Syntax of usage

static declaration

```
static int x;
int foo()
{
  static int foo_invoke=0;
  foo_invoke ++;
  return foo_invoke;
}
```

- A variable can be declared static.
- If the variable is a declared inside a function then all the invocation of the function uses the same variable.
- If it is declared outside every function then it is visible only within that file.

The `extern` declaration

Syntax

extern declaration;

The `extern` declaration

Syntax

extern declaration;

- Says that the definition of the variable is somewhere else, possible in a different file.

The `extern` declaration

Syntax

extern declaration;

- Says that the definition of the variable is somewhere else, possible in a different file.
- Generally used only with variable declarations

The `extern` declaration

Syntax

extern declaration;

- Says that the definition of the variable is somewhere else, possible in a different file.
- Generally used only with variable declarations
- With function declaration has no effect, default declaration is extern declaration.

The `extern` declaration

Syntax

extern declaration;

- Says that the definition of the variable is somewhere else, possible in a different file.
- Generally used only with variable declarations
- With function declaration has no effect, default declaration is extern declaration.
- A variable can be declared `extern` many time but defined only once.