

Forum:[Hack Forums](#)

Thread: [How to create HEADER FILE in c++...and can we include our own methods init](#)

dilipkumarthota - May 5, 2011, 7:00 am

hi frndz,

how to create HEADER FILES in c++...if so i wanna put some classes and functions to it..can we do this..?

bsod666 - May 5, 2011, 5:05 pm

A header file is just a C++ source file, like a cpp file, except it has the h (or hpp) extension. The main difference though is that it doesn't get compiled by the compiler, instead it gets placed inside your cpp file that includes it before the cpp gets compiled (part of the preprocessor step). So say I have two files (test.cpp and test.h) like this:

test.h:

```
Code void myfunc(int param);
void myotherfunc(char *param);
```

test.cpp

```
Code #include "test.h"
```

```
int main(int argc, char **argv) {
    myfunc(10);
    myotherfunc("hello");
    return 0;
}
```

Now the first thing the compiler does to build test.cpp is run it through the preprocessor, after that step it will look like this:

test.cpp after preprocessing:

```
Code void myfunc(int param);
void myotherfunc(char *param);
```

```
int main(int argc, char **argv) {
    myfunc(10);
    myotherfunc("hello");
    return 0;
}
```

Now it will be actually compiled. See how that works? Of course if this was to actually compile I'd need another source file with the actual bodies of the functions declared in test.h though, but test.h contains their declarations so other files know how to call them.

dilipkumarthota - May 6, 2011, 8:29 pm

(May 5, 2011 05:05 PM)bsod666 Wrote: [\[CITE\]](#) A header file is just a C++ source file, like a cpp file, except it has the h (or hpp) extension. The main difference though is that it doesn't get compiled by the compiler, instead it gets placed inside your cpp file that includes it before the cpp gets compiled (part of the preprocessor step). So say I have two files (test.cpp and test.h) like this:

test.h:

```
Code void myfunc(int param);
```

```
void myotherfunc(char *param);  
test.cpp  
Code #include "test.h"
```

```
int main(int argc, char **argv) {  
    myfunc(10);  
    myotherfunc("hello");  
    return 0;  
}
```

Now the first thing the compiler does to build test.cpp is run it through the preprocessor, after that step it will look like this:

```
test.cpp after preprocessing:  
Code void myfunc(int param);  
void myotherfunc(char *param);
```

```
int main(int argc, char **argv) {  
    myfunc(10);  
    myotherfunc("hello");  
    return 0;  
}
```

Now it will be actually compiled. See how that works? Of course if this was to actually compile I'd need another source file with the actual bodies of the functions declared in test.h though, but test.h contains their declarations so other files know how to call them.

thanks for the code....really very helpful