



Getting Started with Java on Linux



About Circe

- All USF students have access to a Linux system known as Circe.
- Try logging in with your USF NetID.
 - (Details follow.)
- If unsuccessful, contact the USF IT Help Desk, 974-1222.

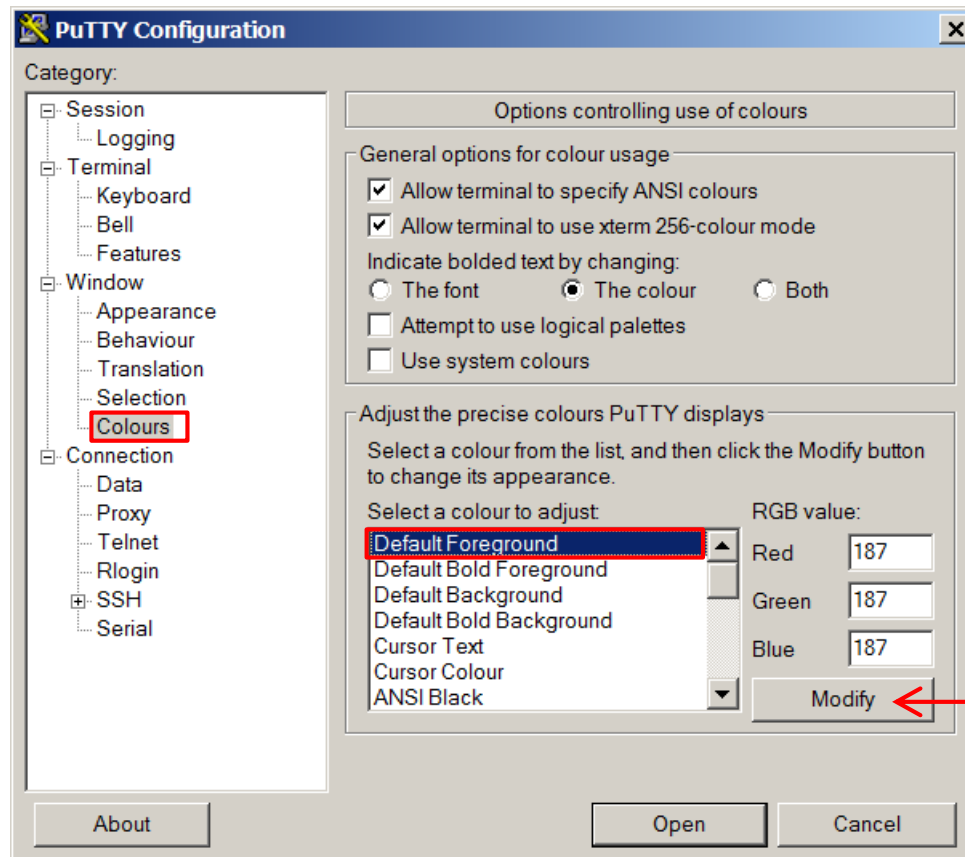


Connecting to Circe

- Use an SSH terminal client program to connect to Circe.
- Recommended client program is PuTTY
 - Can download from <http://www.putty.org/>
- Just click on putty.exe to run it
- PuTTY is available on the ENB 116 computers.
 - Click the Start button and type putty into the search box.

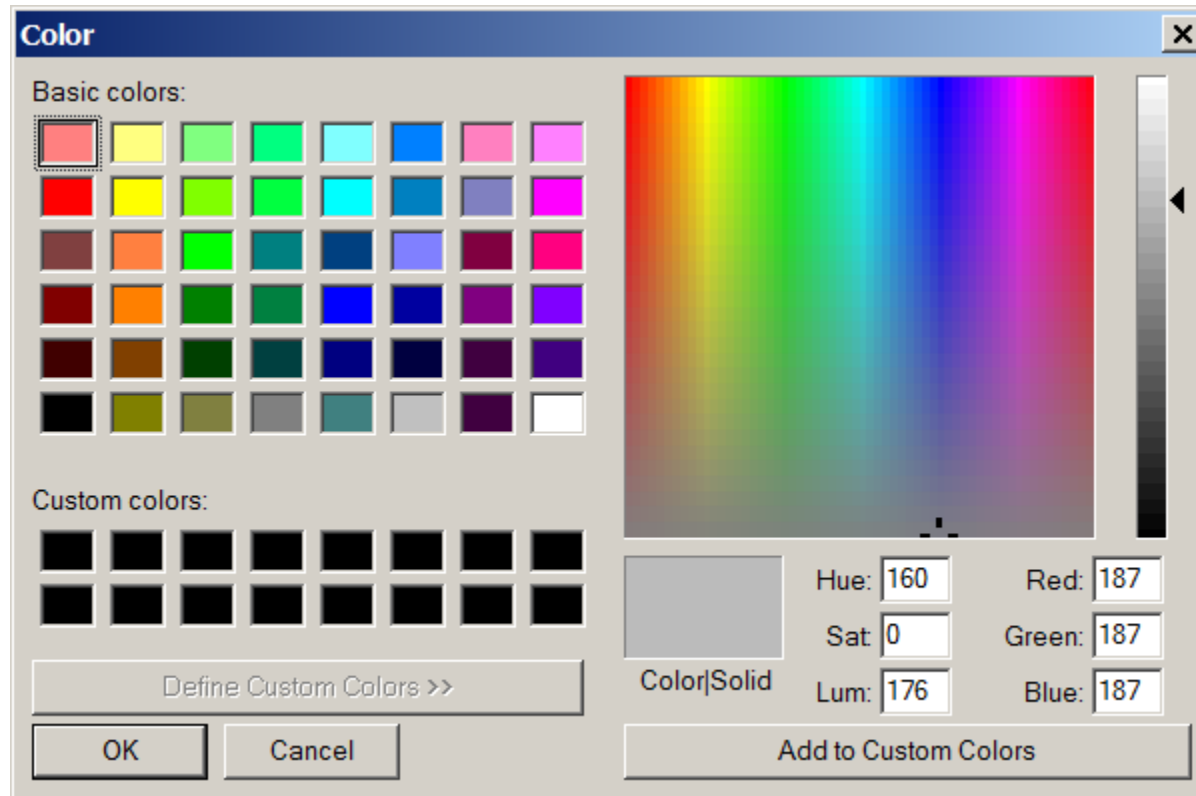
Setting Screen Colors in PuTTY

Not necessary if you like white text on black background.



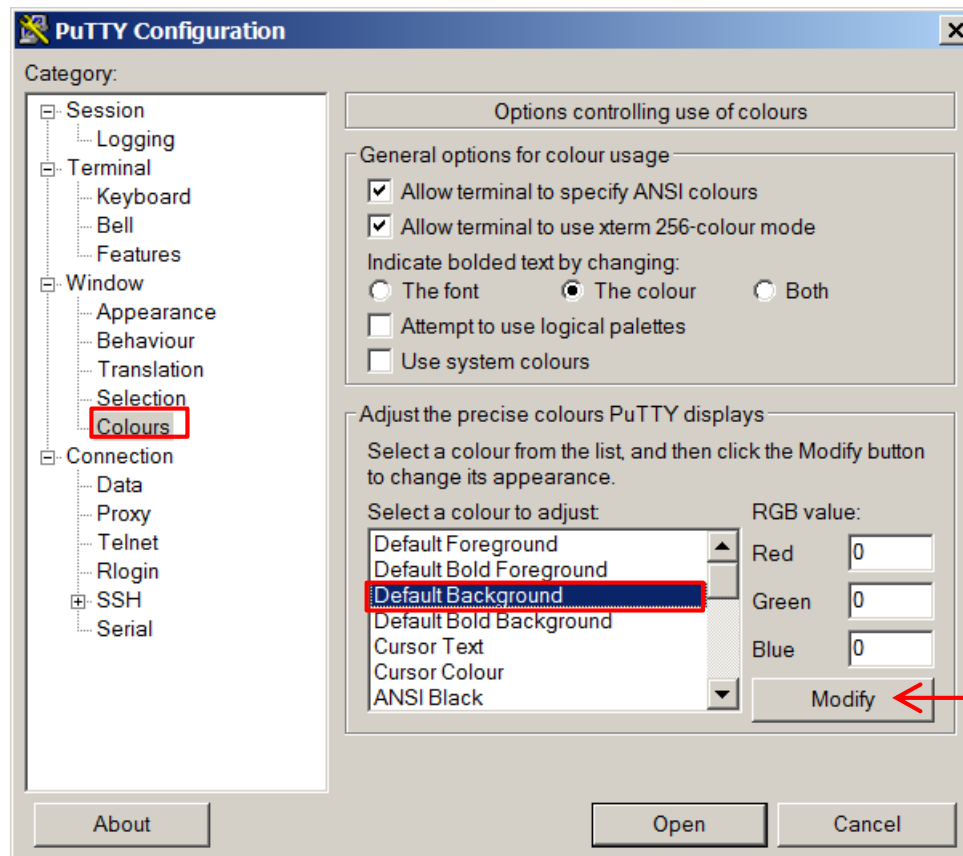
Setting Screen Colors in PuTTY

Click on the color that you want for text.



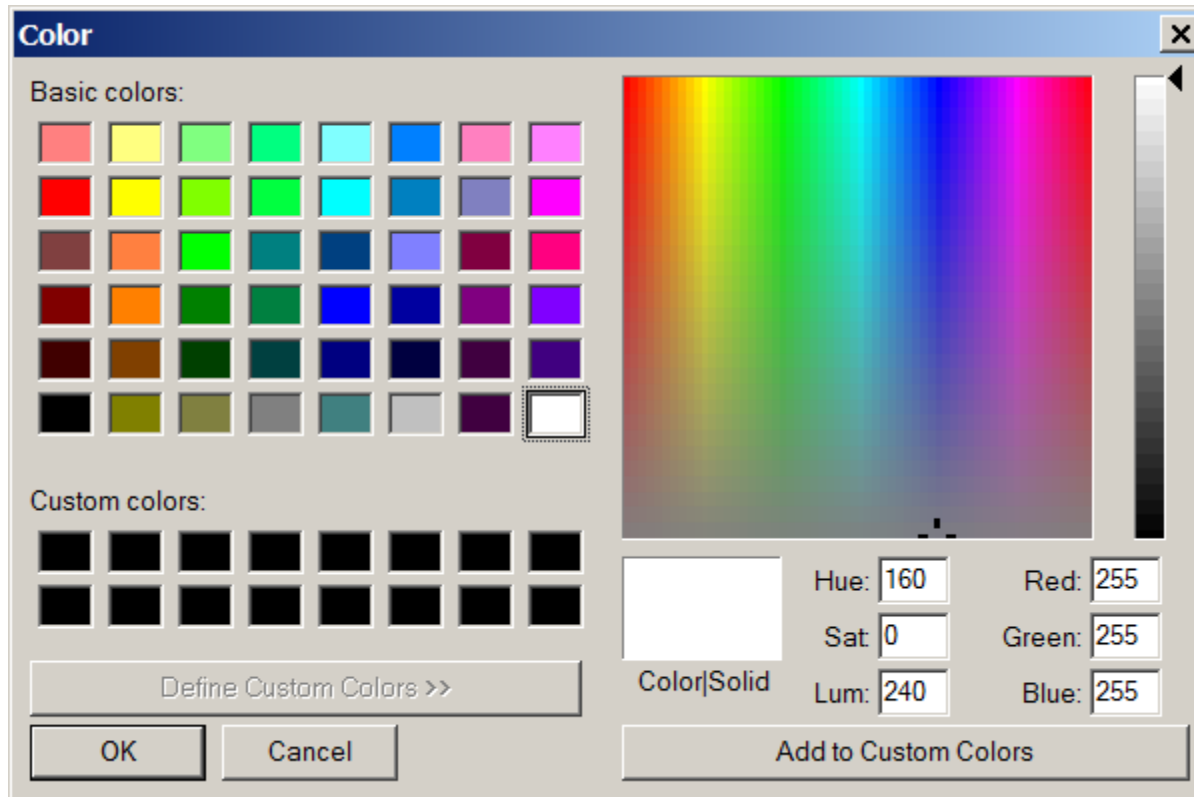
Then click OK

Setting Screen Colors in PuTTY



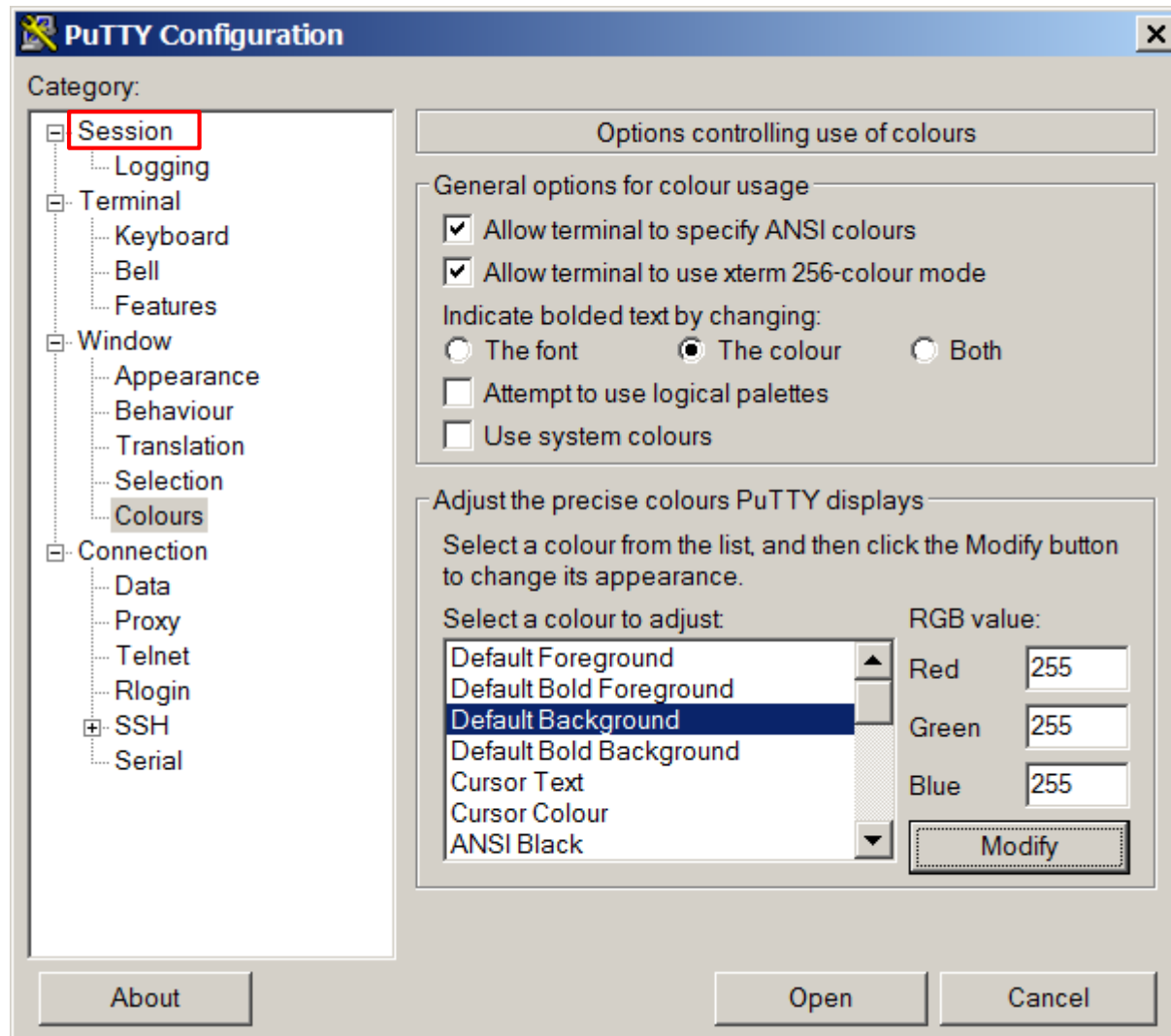
Setting Screen Colors in PuTTY

Click on the color that you want for background



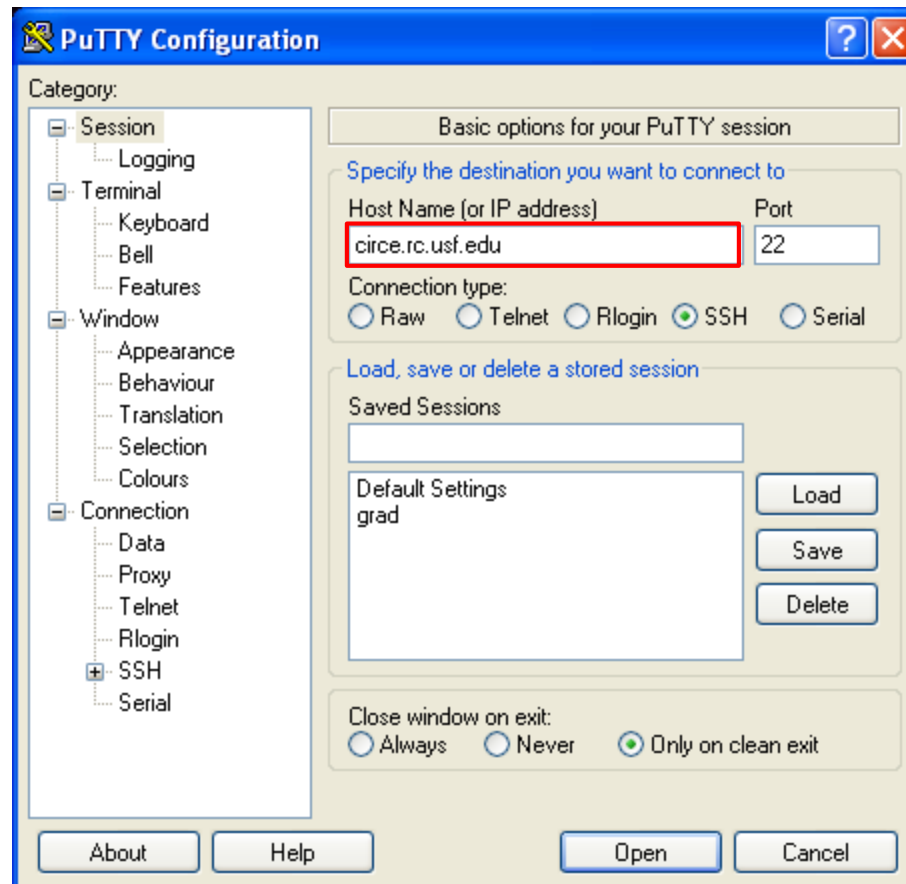
Then click OK

Setting Screen Colors in PuTTY



Click on Session

Connecting with PuTTY



Click Open

Using Circe

```
turnerr@login0:-
login as: turnerr Your Net ID

University of South Florida Central Instructional and Research
Computing Environment (CIRCE)

WARNING: UNAUTHORIZED ACCESS TO THIS SYSTEM IS PROHIBITED BY LAW

turnerr@circe.rc.usf.edu's password: Your Net ID Password
Last login: Sun Nov 28 20:51:29 2010 from 72.187.18.171

Documentation: https://rc.usf.edu/trac/doc
Support: help@usf.edu
Phone: 813-974-1222

To change your password or to recover a lost password, please visit
https://netid.usf.edu

Please report any and all problems to the Helpdesk at help@usf.edu so we
can track your incident!

[turnerr@login0 ~]$
```

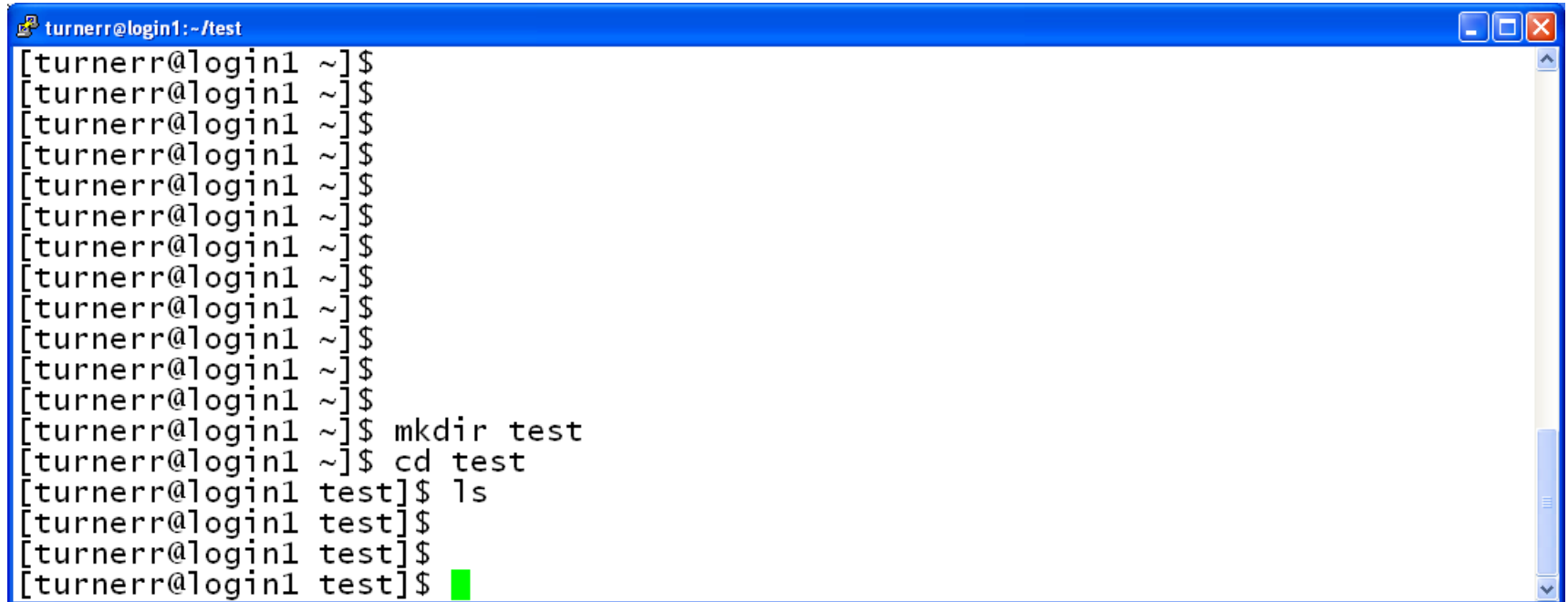
■ Command Prompt



Using Circe

- Linux commands are different from what you saw for the Windows command line.
- Note the commands closely on the following slides.

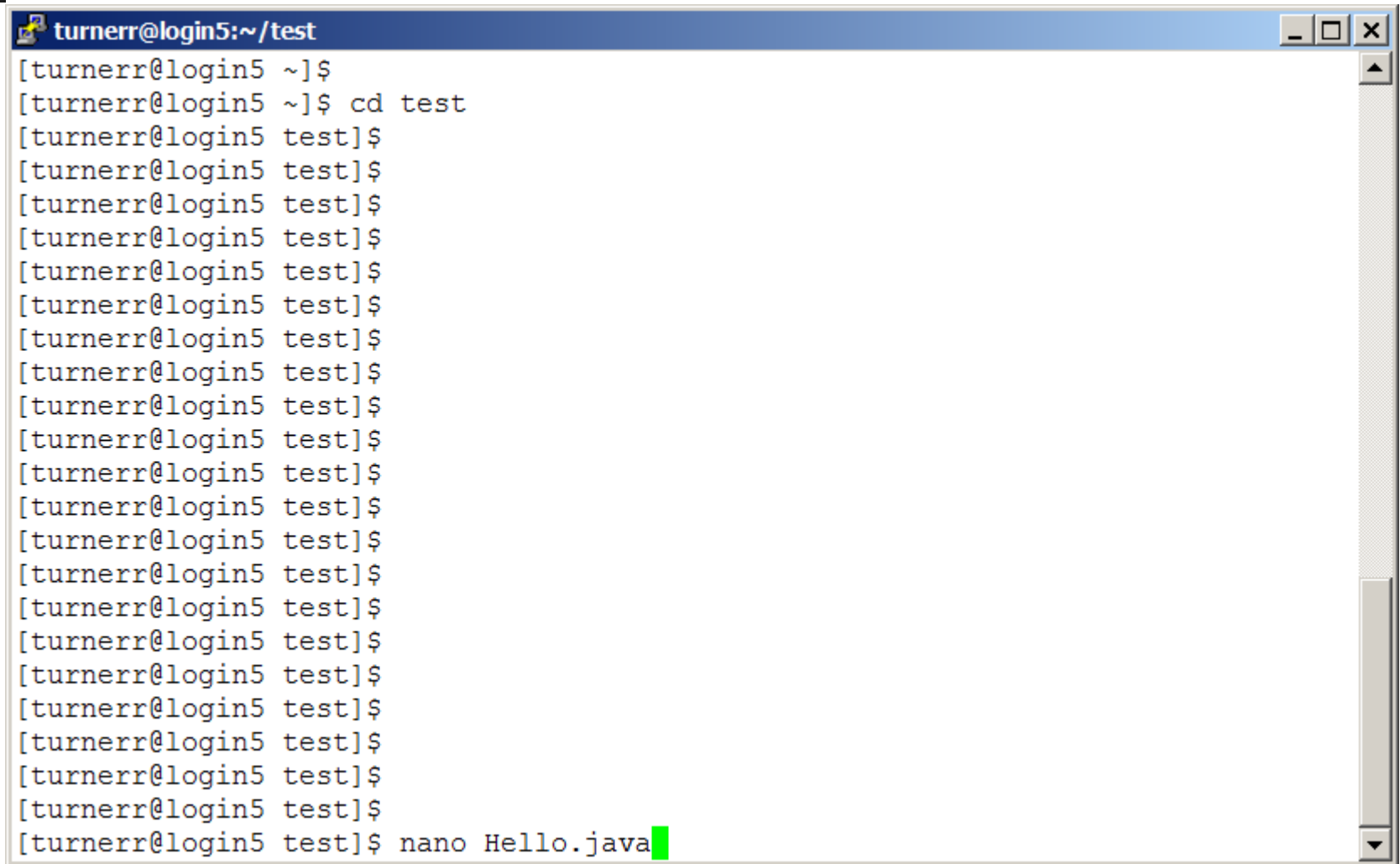
Create a Test Directory

A terminal window titled 'turnerr@login1: ~/test' with a blue header bar. The window contains a series of shell prompts and commands. The first 12 prompts are '[turnerr@login1 ~]\$' and the next 3 are '[turnerr@login1 test]\$'. The commands entered are 'mkdir test', 'cd test', and 'ls'. A green cursor is visible at the end of the last prompt.

```
turnerr@login1: ~/test
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$ mkdir test
[turnerr@login1 ~]$ cd test
[turnerr@login1 test]$ ls
[turnerr@login1 test]$
[turnerr@login1 test]$
[turnerr@login1 test]$
```

mkdir	Create a directory
cd	Change Directory
ls	List the contents of the current directory

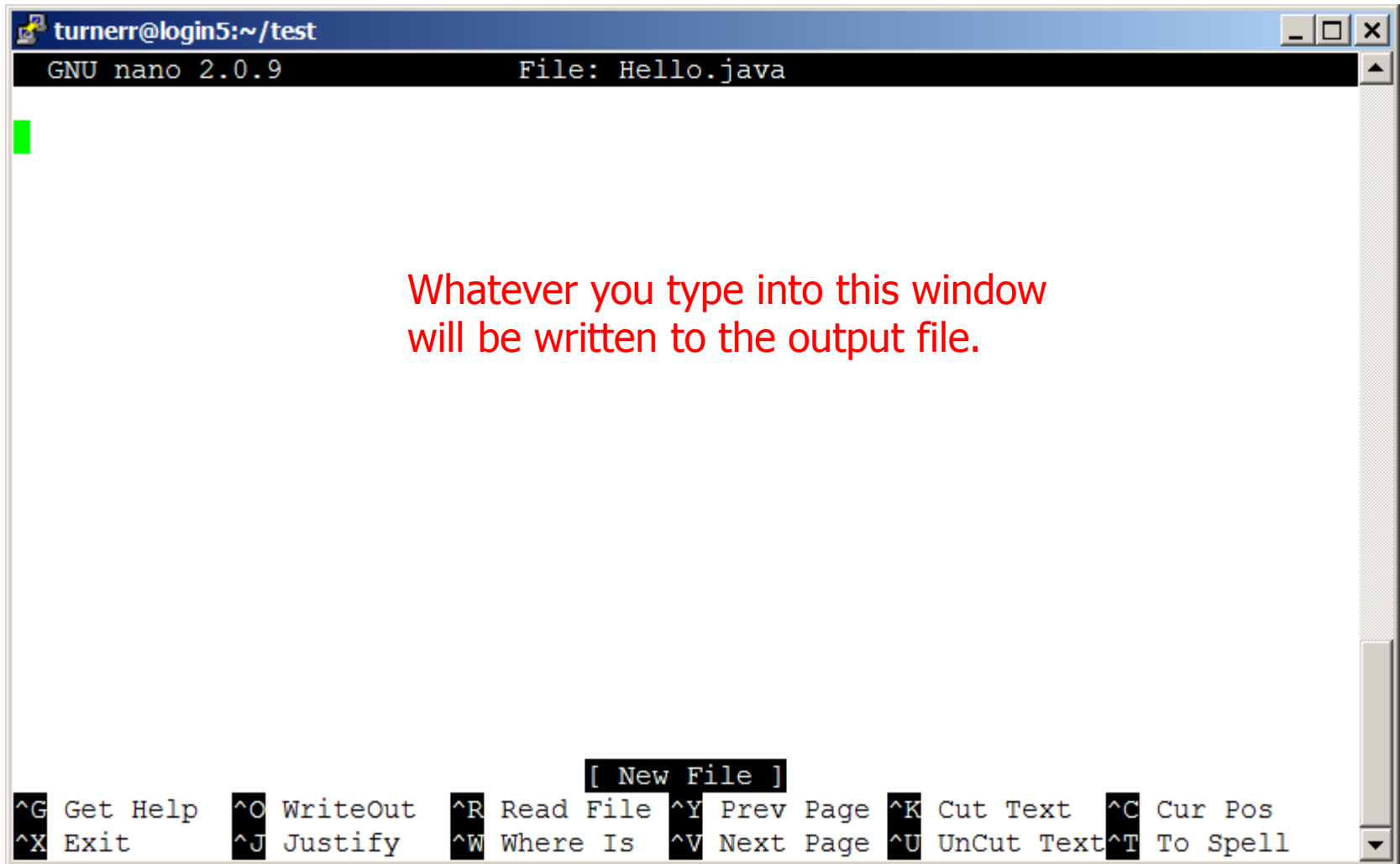
Creating a Source File on Circe



```
turnerr@login5:~/test
[turnerr@login5 ~]$
[turnerr@login5 ~]$ cd test
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$
[turnerr@login5 test]$ nano Hello.java
```

nano is a very simple text editor.

Creating a Source File on Circe



```
turnerr@login5:~/test
GNU nano 2.0.9          File: Hello.java

[ New File ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Whatever you type into this window
will be written to the output file.

Creating a Source File on Circe



```
turnerr@login5:~
GNU nano 2.0.9      File: Hello.java      Modified

/*
 * This Java program simply outputs "Hello, World!" to the screen.
 */
class Hello
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Press Ctrl-o to write out the file.

Creating a Source File on Circe



```
turnerr@login5:~
GNU nano 2.0.9      File: Hello.java      Modified

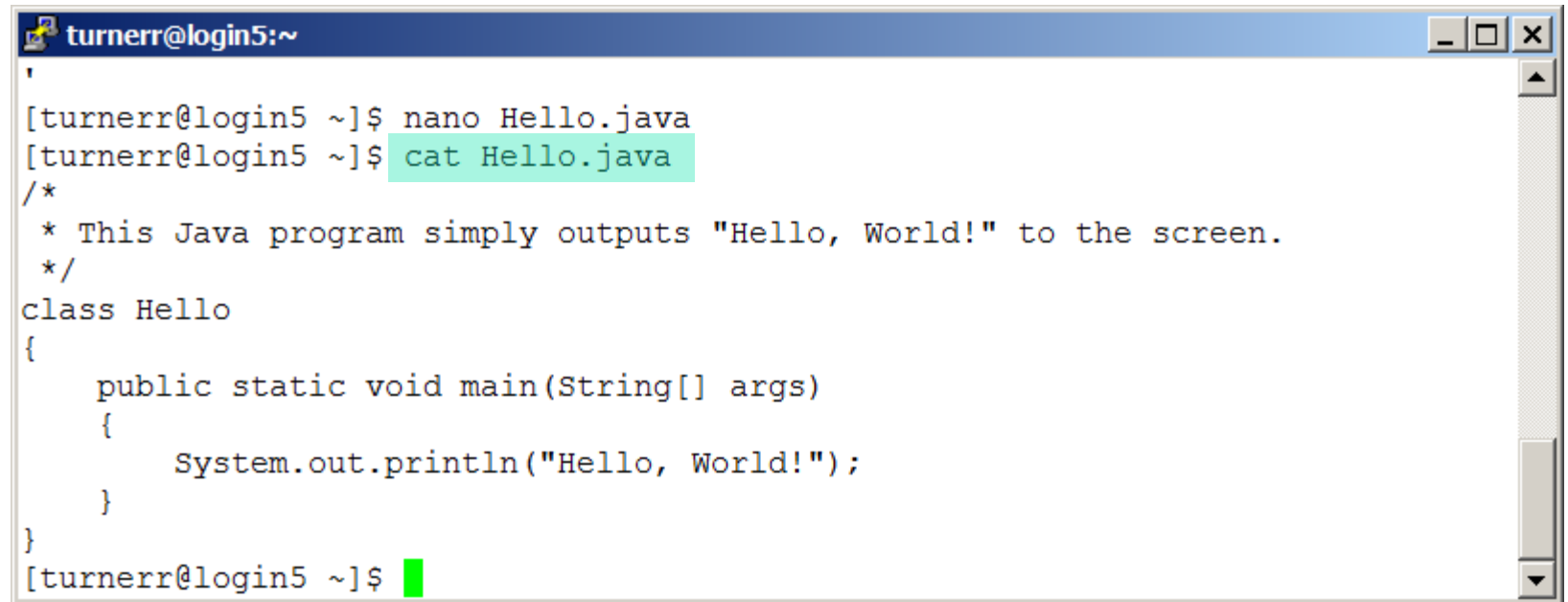
/*
 * This Java program simply outputs "Hello, World!" to the screen.
 */
class Hello
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}

File Name to Write: Hello.java
^G Get Help      ^T To Files      M-M Mac Format   M-P Prepend
^C Cancel        M-D DOS Format   M-A Append       M-B Backup File
```

Press Enter to write contents of window to file
hello.java in the current directory.

The press Control-x to exit from nano.

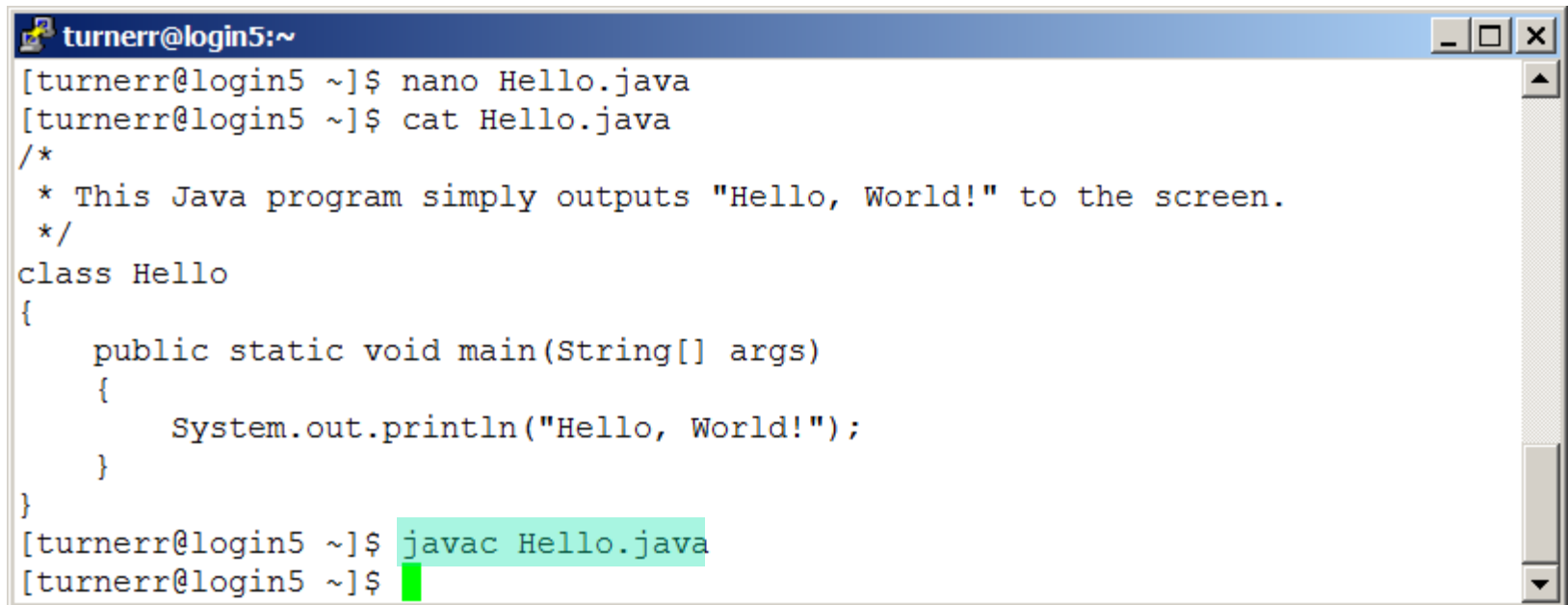
View the Source File



```
turnerr@login5:~  
'  
[turnerr@login5 ~]$ nano Hello.java  
[turnerr@login5 ~]$ cat Hello.java  
/*  
 * This Java program simply outputs "Hello, World!" to the screen.  
 */  
class Hello  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello, World!");  
    }  
}  
[turnerr@login5 ~]$
```

cat Output the contents of a file to the screen

Compile the Program

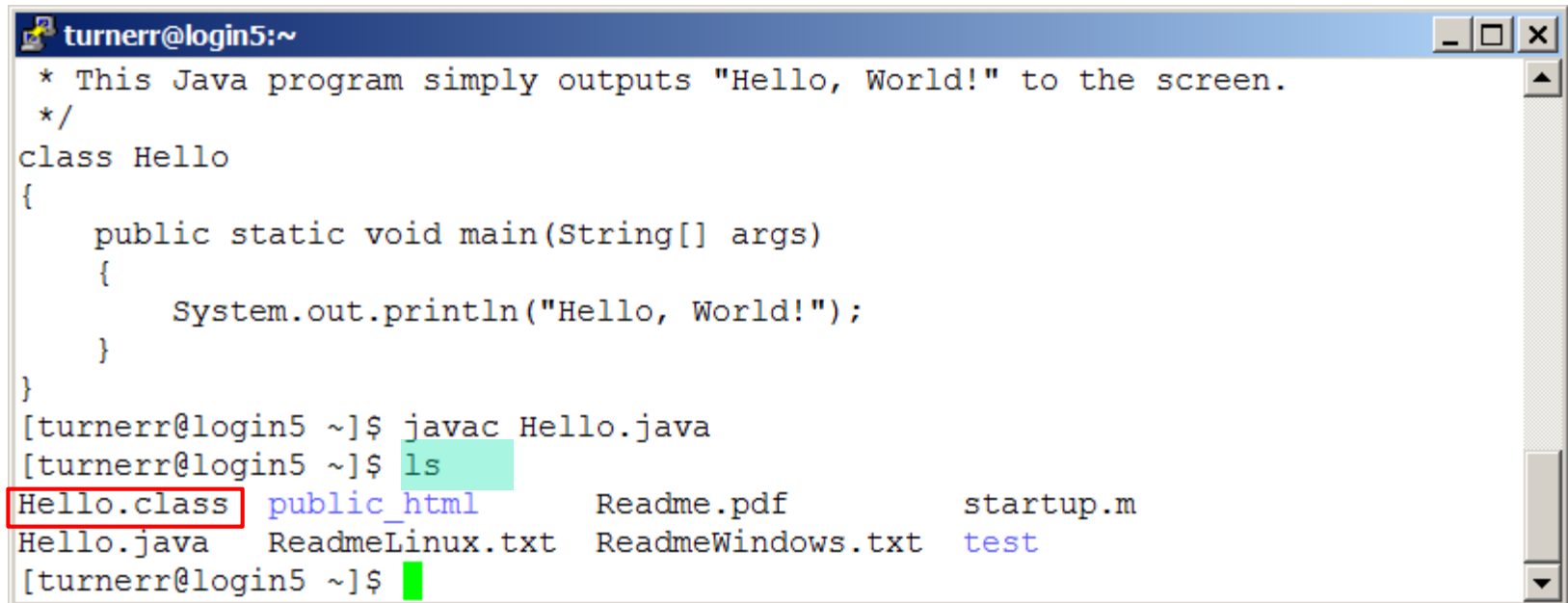


```
turnerr@login5:~  
[turnerr@login5 ~]$ nano Hello.java  
[turnerr@login5 ~]$ cat Hello.java  
/*  
 * This Java program simply outputs "Hello, World!" to the screen.  
 */  
class Hello  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello, World!");  
    }  
}  
[turnerr@login5 ~]$ javac Hello.java  
[turnerr@login5 ~]$
```

No output means that your compilation was successful.

List the directory

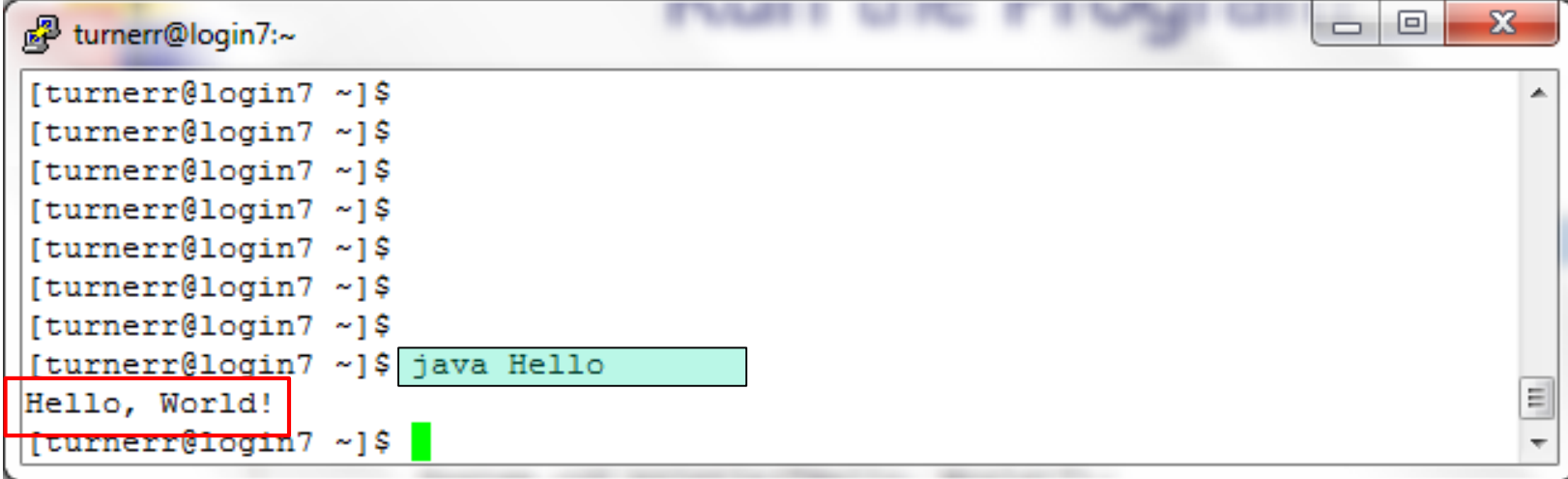
Here is our Java bytecode file.

A terminal window titled 'turnerr@login5:~' with standard window controls. It contains a Java comment, a class definition for 'Hello', the compilation command 'javac Hello.java', and the execution of 'ls' which lists files including 'Hello.class' (highlighted with a red box), 'public_html', 'Readme.pdf', 'startup.m', 'Hello.java', 'ReadmeLinux.txt', 'ReadmeWindows.txt', and 'test'.

```
turnerr@login5:~
* This Java program simply outputs "Hello, World!" to the screen.
*/
class Hello
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}
[turnerr@login5 ~]$ javac Hello.java
[turnerr@login5 ~]$ ls
Hello.class  public_html  Readme.pdf  startup.m
Hello.java   ReadmeLinux.txt  ReadmeWindows.txt  test
[turnerr@login5 ~]$
```

Hello.class was written by the Java compiler.

Run the Program



A terminal window titled 'turnerr@login7:~' with standard window controls (minimize, maximize, close). The terminal shows a series of empty prompts '[turnerr@login7 ~]\$'. The command 'java Hello' is entered and highlighted with a light blue selection box. The output 'Hello, World!' is displayed on the next line and highlighted with a red rectangular box. A green cursor is visible on the line following the output.

```
turnerr@login7:~  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$  
[turnerr@login7 ~]$ java Hello  
Hello, World!  
[turnerr@login7 ~]$
```

Our
program's
output