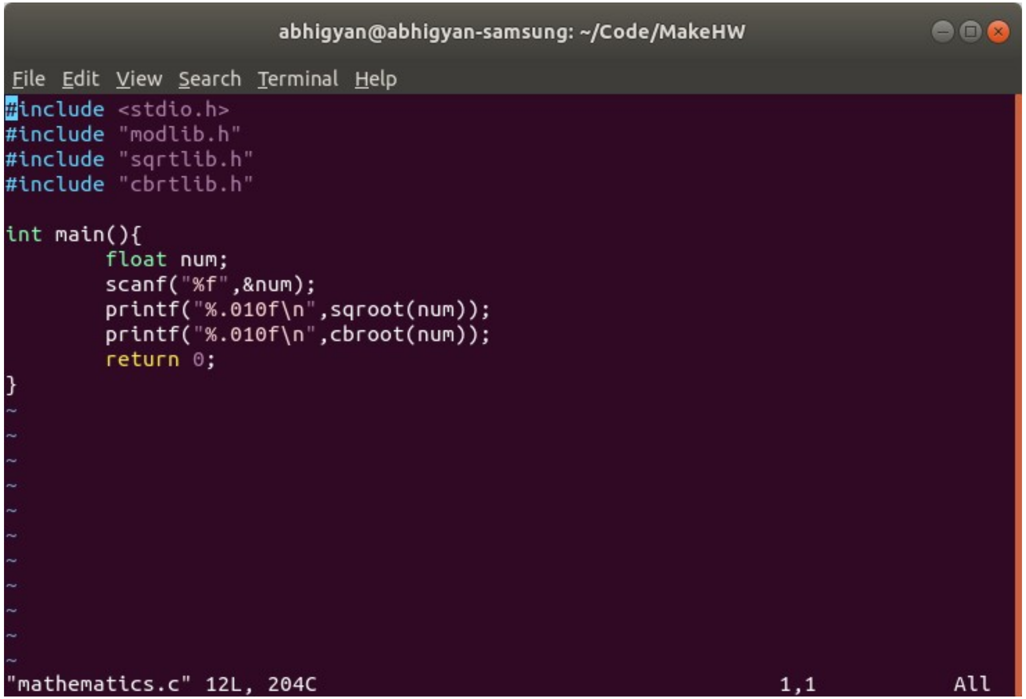
# Homework 5

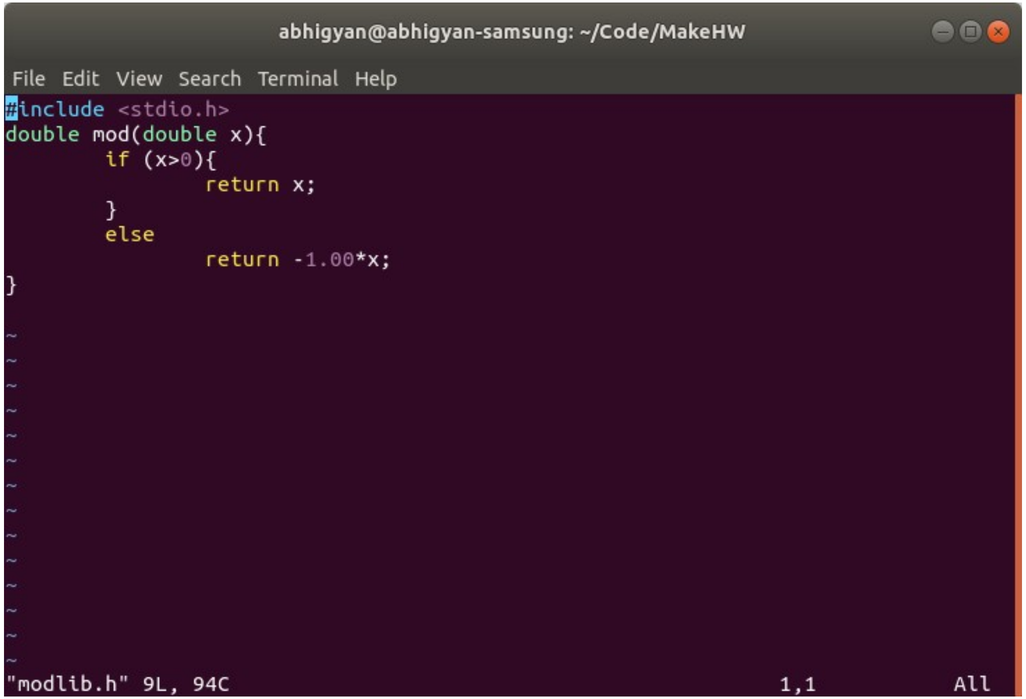
Abhigyan Chattopadhyay
ME19B001

# Homework – Session 10

# 10.1. Take one of your old codes, split the code into separate files, one for each function. Create a makefile and test the recompilation.


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

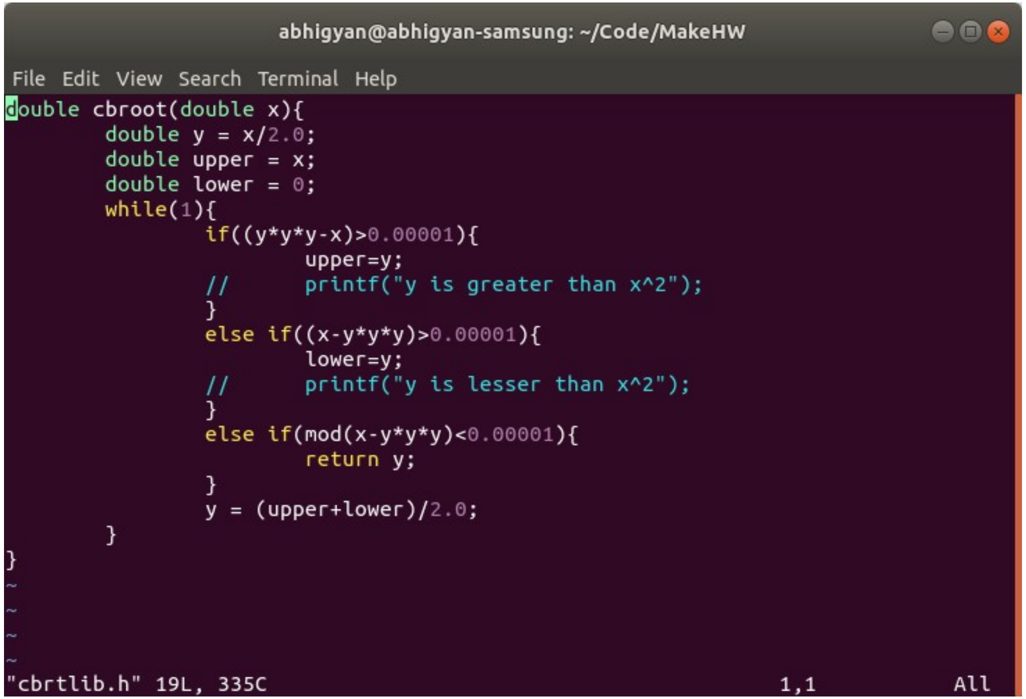
```
#include <stdio.h>
#include "modlib.h"
#include "sqrtlib.h"
#include "cbrtlib.h"

int main(){
        float num;
        scanf("%f",&num);
        printf("%.010f\n",sqroot(num));
        printf("%.010f\n",cbroot(num));
        return 0;
}
~
~
~
~
~
~
~
~
~
~
~
"mathematics.c" 12L, 204C                                 1,1           All
```


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

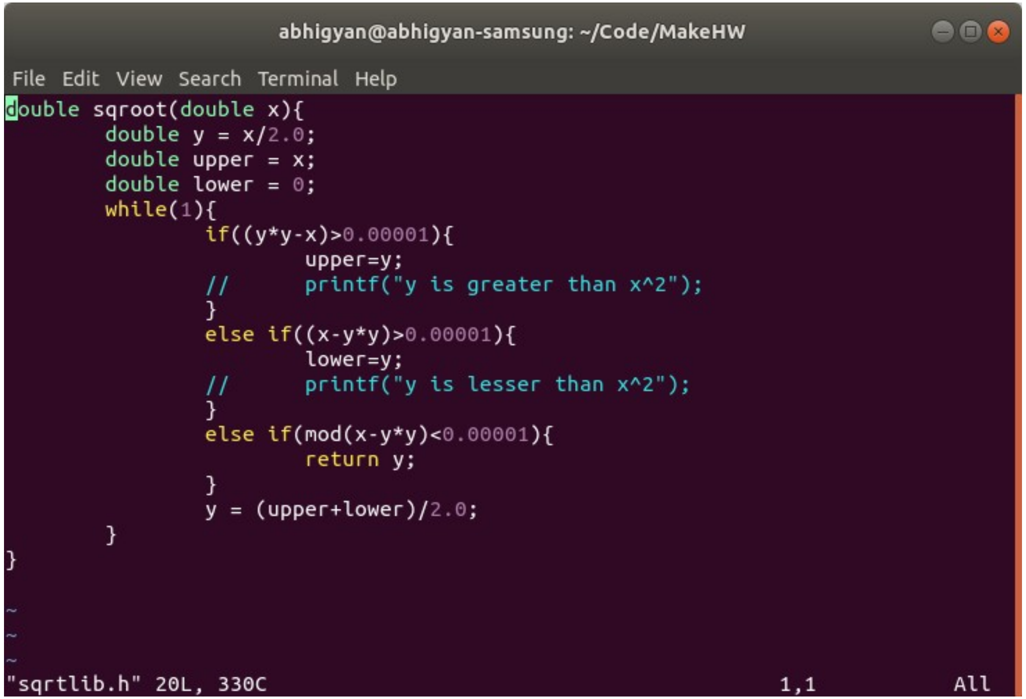
```
#include <stdio.h>
double mod(double x){
        if (x>0){
                return x;
        }
        else
                return -1.00*x;
}

~
~
~
~
~
~
~
~
~
~
~
~
~
~
"modlib.h" 9L, 94C                                        1,1           All
```

For C Language

# 10.1. Take one of your old codes, split the code into separate files, one for each function. Create a makefile and test the recompilation.


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

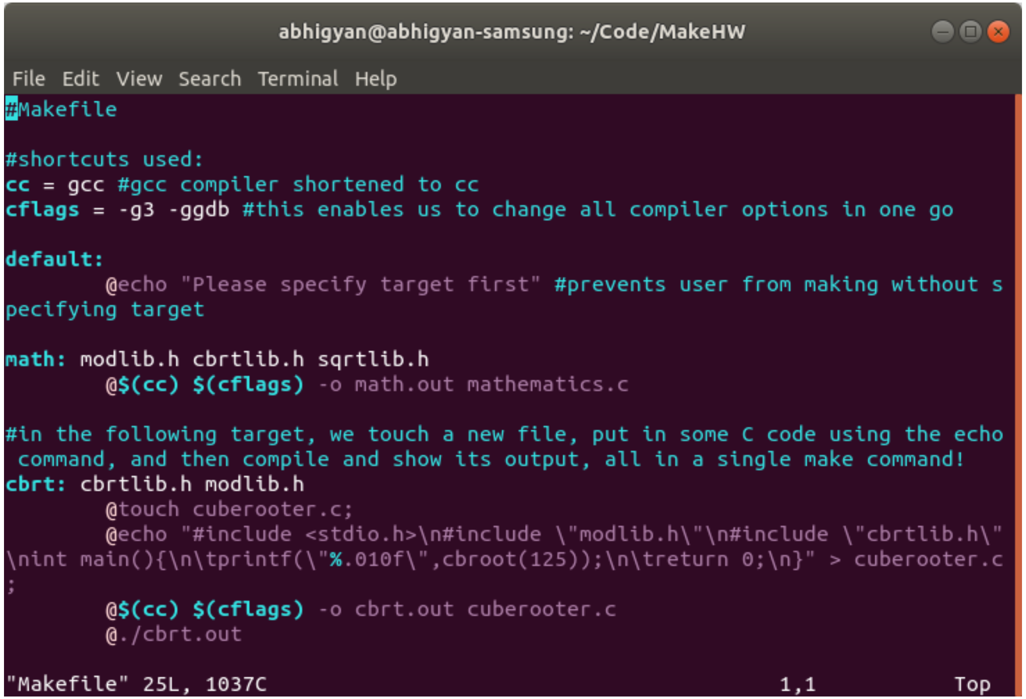
```
double cbroot(double x){
        double y = x/2.0;
        double upper = x;
        double lower = 0;
        while(1){
                if((y*y*y-x)>0.00001){
                        upper=y;
                //      printf("y is greater than x^2");
                }
                else if((x-y*y*y)>0.00001){
                        lower=y;
                //      printf("y is lesser than x^2");
                }
                else if(mod(x-y*y*y)<0.00001){
                        return y;
                }
                y = (upper+lower)/2.0;
        }
}
~
~
~
~
"cbrtlib.h" 19L, 335C                                   1,1           All
```


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

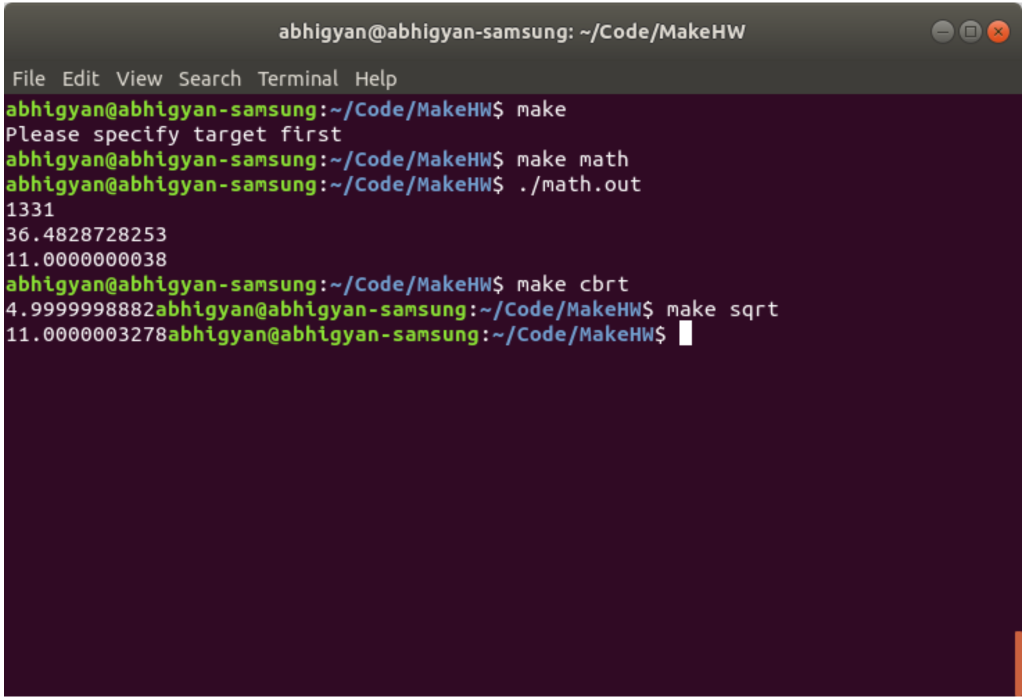
```
double sqroot(double x){
        double y = x/2.0;
        double upper = x;
        double lower = 0;
        while(1){
                if((y*y-x)>0.00001){
                        upper=y;
                //      printf("y is greater than x^2");
                }
                else if((x-y*y)>0.00001){
                        lower=y;
                //      printf("y is lesser than x^2");
                }
                else if(mod(x-y*y)<0.00001){
                        return y;
                }
                y = (upper+lower)/2.0;
        }
}

~
~
~
"sqrtlib.h" 20L, 330C                                   1,1           All
```

For C Language

# 10.1. Take one of your old codes, split the code into separate files, one for each function. Create a makefile and test the recompilation.


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

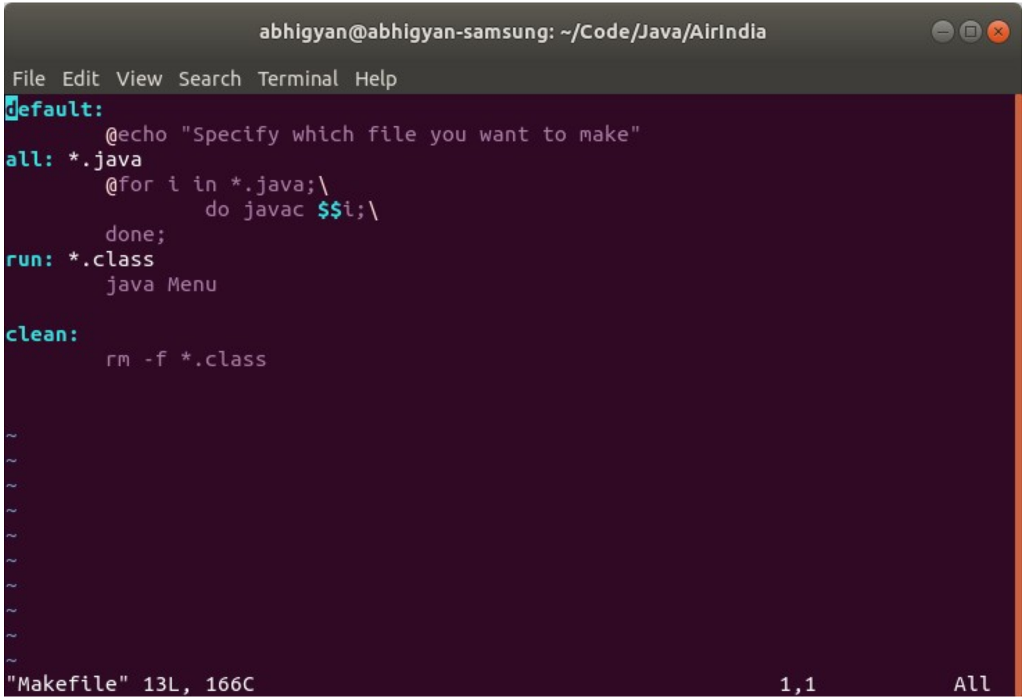
```
#Makefile

#shortcuts used:
cc = gcc #gcc compiler shortened to cc
cflags = -g3 -ggdb #this enables us to change all compiler options in one go

default:
        @echo "Please specify target first" #prevents user from making without s
pecifying target

math: modlib.h cbrtlib.h sqrtlib.h
        @$(cc) $(cflags) -o math.out mathematics.c

#in the following target, we touch a new file, put in some C code using the echo
 command, and then compile and show its output, all in a single make command!
cbrt: cbrtlib.h modlib.h
        @touch cuberooter.c;
        @echo "#include <stdio.h>\n#include \"modlib.h\"\n#include \"cbrtlib.h\"
\nint main(){\n\tprintf(\"%.010f\",cbroot(125));\n\treturn 0;\n}" > cuberooter.c
;
        @$(cc) $(cflags) -o cbrt.out cuberooter.c
        @./cbrt.out

"Makefile" 25L, 1037C                                         1,1           Top
```


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

```
abhigyan@abhigyan-samsung:~/Code/MakeHW$ make
Please specify target first
abhigyan@abhigyan-samsung:~/Code/MakeHW$ make math
abhigyan@abhigyan-samsung:~/Code/MakeHW$ ./math.out
1331
36.4828728253
11.0000000038
abhigyan@abhigyan-samsung:~/Code/MakeHW$ make cbrt
4.9999998882abhigyan@abhigyan-samsung:~/Code/MakeHW$ make sqrt
11.0000003278abhigyan@abhigyan-samsung:~/Code/MakeHW$
```

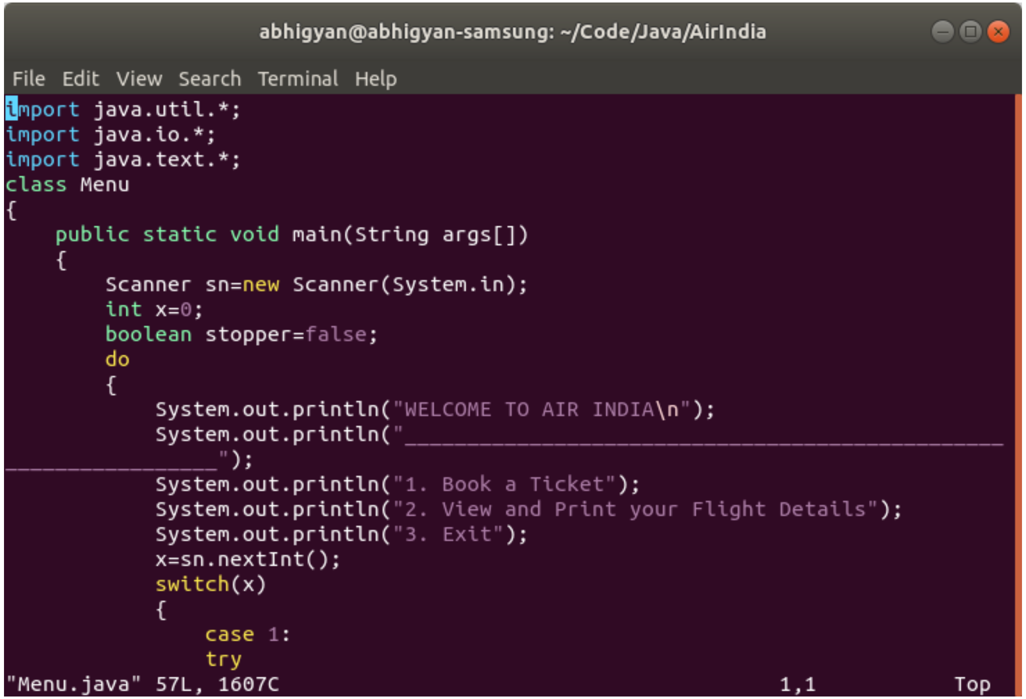
For C Language

# 10.1. Take one of your old codes, split the code into separate files, one for each function. Create a makefile and test the recompilation.



abhigyan@abhigyan-samsung: ~/Code/Java/AirIndia
File Edit View Search Terminal Help

```
default:
        @echo "Specify which file you want to make"
all: *.java
        @for i in *.java;\
                do javac $$i;\
        done;
run: *.class
        java Menu

clean:
        rm -f *.class



~
~
~
~
~
~
~
~
~
~
"Makefile" 13L, 166C                                          1,1           All
```

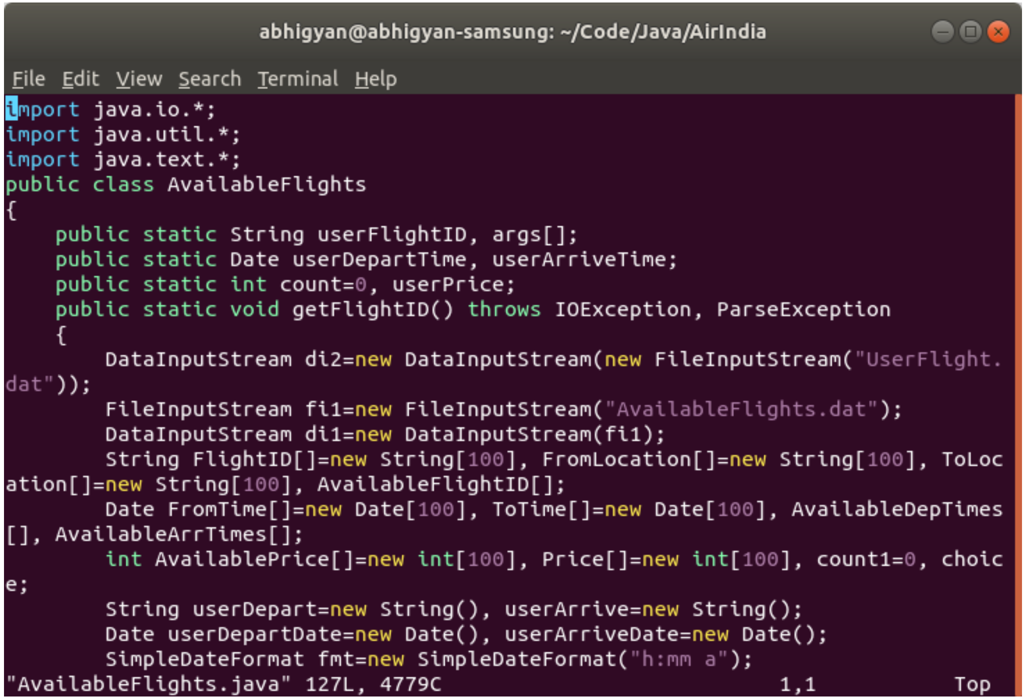


abhigyan@abhigyan-samsung: ~/Code/Java/AirIndia
File Edit View Search Terminal Help

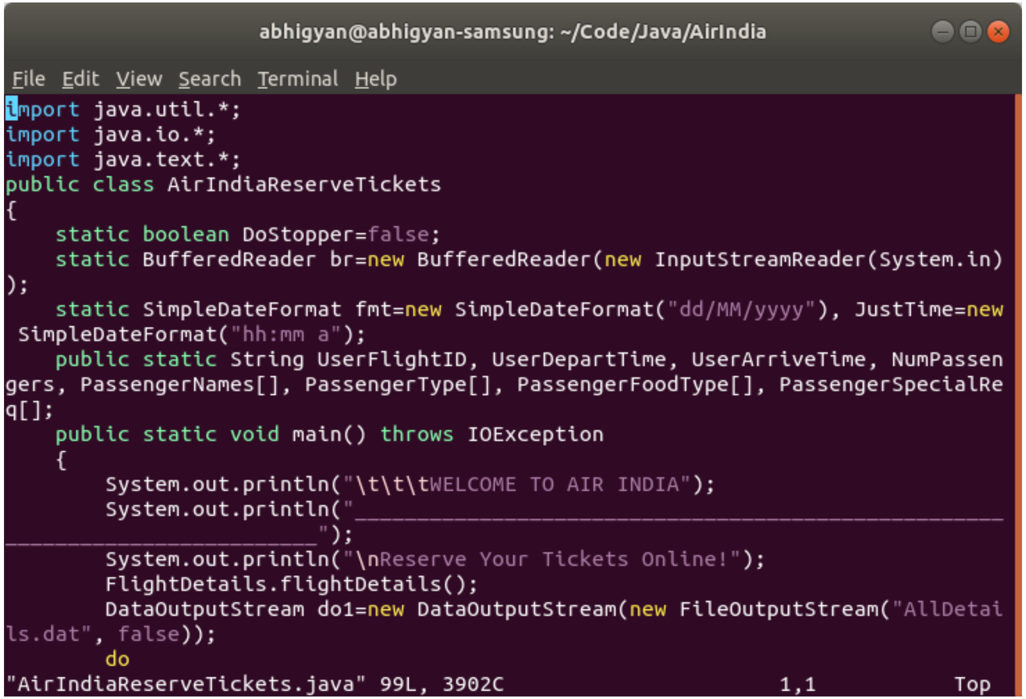
```
import java.util.*;
import java.io.*;
import java.text.*;
class Menu
{
    public static void main(String args[])
    {
        Scanner sn=new Scanner(System.in);
        int x=0;
        boolean stopper=false;
        do
        {
            System.out.println("WELCOME TO AIR INDIA\n");
            System.out.println("________________________________________________
_________________");
            System.out.println("1. Book a Ticket");
            System.out.println("2. View and Print your Flight Details");
            System.out.println("3. Exit");
            x=sn.nextInt();
            switch(x)
            {
                case 1:
                try
"Menu.java" 57L, 1607C                                        1,1           Top
```

For Java

# 10.1. Take one of your old codes, split the code into separate files, one for each function. Create a makefile and test the recompilation.


abhigyan@abhigyan-samsung: ~/Code/Java/AirIndia
File Edit View Search Terminal Help

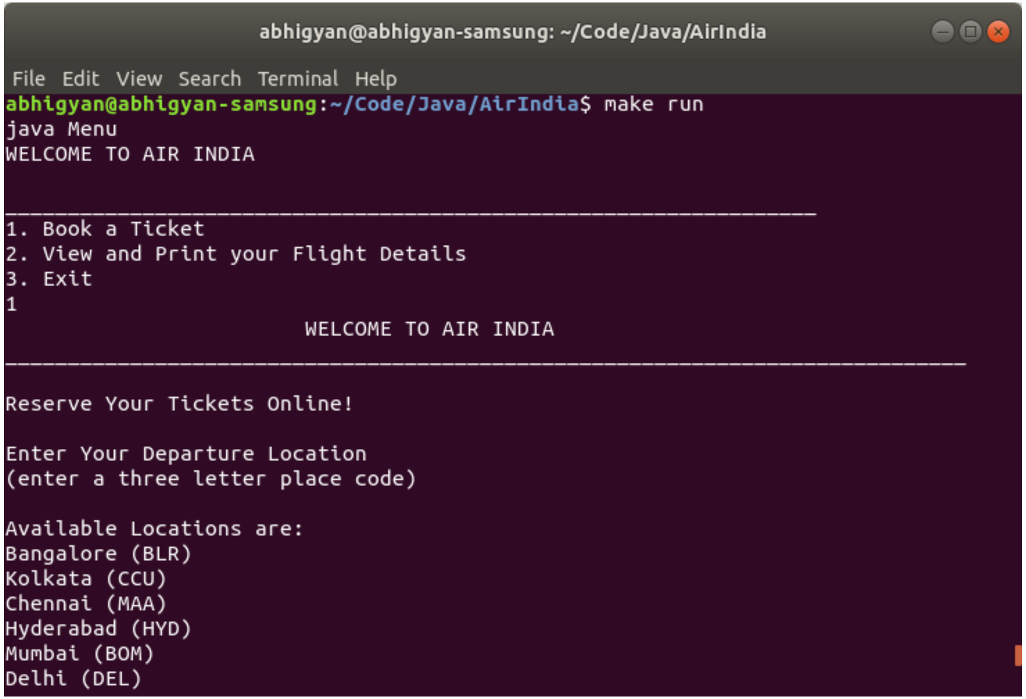
```
import java.io.*;
import java.util.*;
import java.text.*;
public class AvailableFlights
{
    public static String userFlightID, args[];
    public static Date userDepartTime, userArriveTime;
    public static int count=0, userPrice;
    public static void getFlightID() throws IOException, ParseException
    {
        DataInputStream di2=new DataInputStream(new FileInputStream("UserFlight.
dat"));
        FileInputStream fi1=new FileInputStream("AvailableFlights.dat");
        DataInputStream di1=new DataInputStream(fi1);
        String FlightID[]=new String[100], FromLocation[]=new String[100], ToLoc
ation[]=new String[100], AvailableFlightID[];
        Date FromTime[]=new Date[100], ToTime[]=new Date[100], AvailableDepTimes
[], AvailableArrTimes[];
        int AvailablePrice[]=new int[100], Price[]=new int[100], count1=0, choic
e;
        String userDepart=new String(), userArrive=new String();
        Date userDepartDate=new Date(), userArriveDate=new Date();
        SimpleDateFormat fmt=new SimpleDateFormat("h:mm a");
"AvailableFlights.java" 127L, 4779C                           1,1           Top
```


abhigyan@abhigyan-samsung: ~/Code/Java/AirIndia
File Edit View Search Terminal Help

```
import java.util.*;
import java.io.*;
import java.text.*;
public class AirIndiaReserveTickets
{
    static boolean DoStopper=false;
    static BufferedReader br=new BufferedReader(new InputStreamReader(System.in)
);
    static SimpleDateFormat fmt=new SimpleDateFormat("dd/MM/yyyy"), JustTime=new
 SimpleDateFormat("hh:mm a");
    public static String UserFlightID, UserDepartTime, UserArriveTime, NumPassen
gers, PassengerNames[], PassengerType[], PassengerFoodType[], PassengerSpecialRe
q[];
    public static void main() throws IOException
    {
        System.out.println("\t\t\tWELCOME TO AIR INDIA");
        System.out.println("____________________________________________________
_________________________");
        System.out.println("\nReserve Your Tickets Online!");
        FlightDetails.flightDetails();
        DataOutputStream do1=new DataOutputStream(new FileOutputStream("AllDetai
ls.dat", false));
        do
"AirIndiaReserveTickets.java" 99L, 3902C                      1,1           Top
```

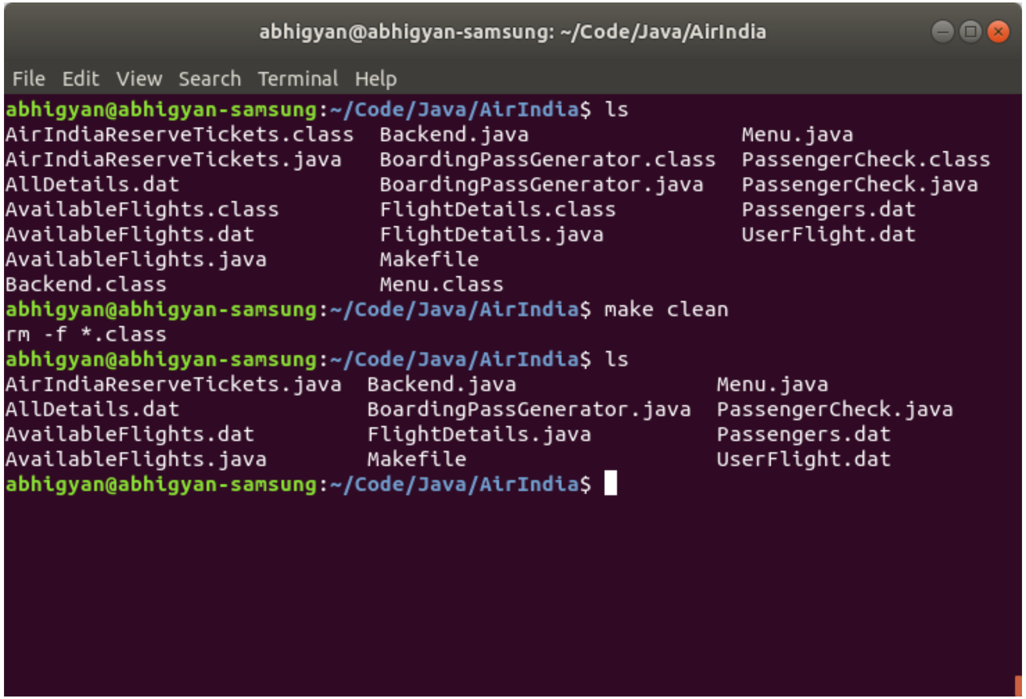
For Java

# 10.1. Take one of your old codes, split the code into separate files, one for each function. Create a makefile and test the recompilation.



abhigyan@abhigyan-samsung: ~/Code/Java/AirIndia
File Edit View Search Terminal Help

```
abhigyan@abhigyan-samsung:~/Code/Java/AirIndia$ make run
java Menu
WELCOME TO AIR INDIA


_________________________________________________________________
1. Book a Ticket
2. View and Print your Flight Details
3. Exit
1
                        WELCOME TO AIR INDIA

_____________________________________________________________________________

Reserve Your Tickets Online!

Enter Your Departure Location
(enter a three letter place code)

Available Locations are:
Bangalore (BLR)
Kolkata (CCU)
Chennai (MAA)
Hyderabad (HYD)
Mumbai (BOM)
Delhi (DEL)
```



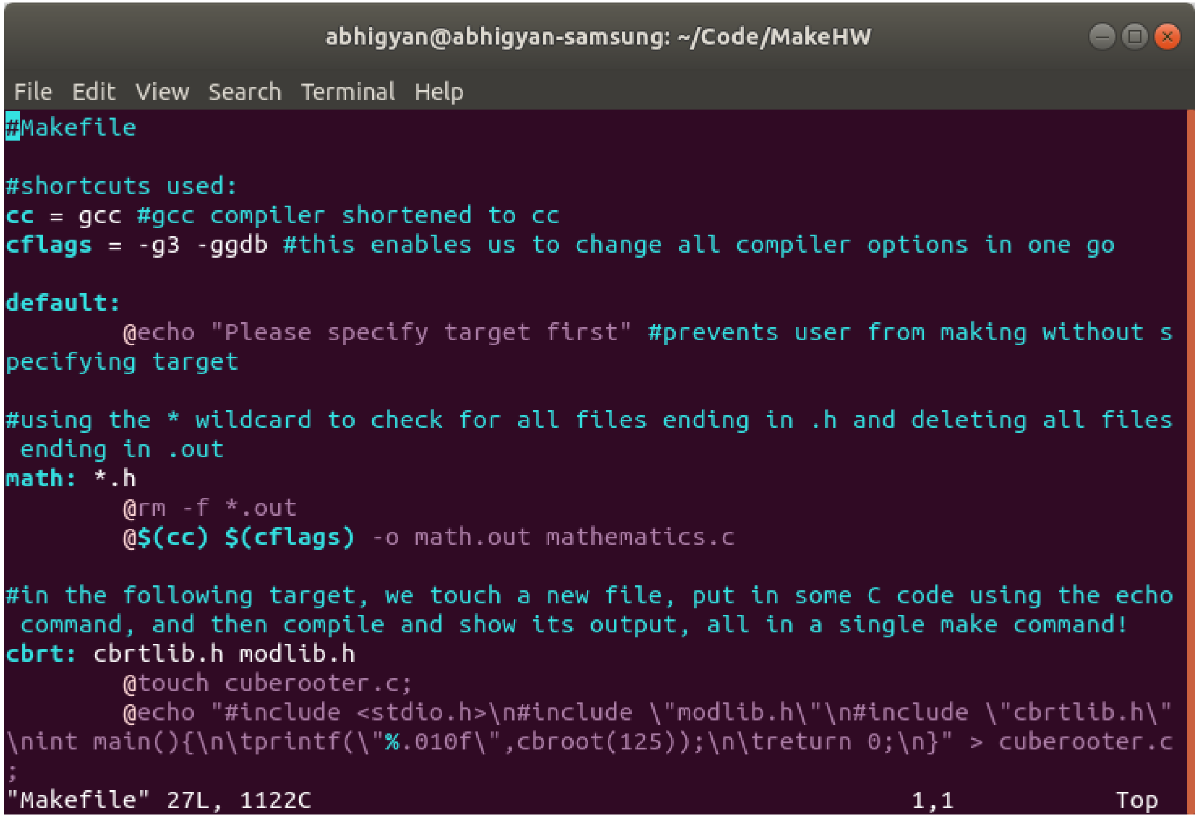
abhigyan@abhigyan-samsung: ~/Code/Java/AirIndia
File Edit View Search Terminal Help

```
abhigyan@abhigyan-samsung:~/Code/Java/AirIndia$ ls
AirIndiaReserveTickets.class  Backend.java                 Menu.java
AirIndiaReserveTickets.java   BoardingPassGenerator.class  PassengerCheck.class
AllDetails.dat                BoardingPassGenerator.java   PassengerCheck.java
AvailableFlights.class        FlightDetails.class          Passengers.dat
AvailableFlights.dat          FlightDetails.java           UserFlight.dat
AvailableFlights.java         Makefile
Backend.class                 Menu.class
abhigyan@abhigyan-samsung:~/Code/Java/AirIndia$ make clean
rm -f *.class
abhigyan@abhigyan-samsung:~/Code/Java/AirIndia$ ls
AirIndiaReserveTickets.java  Backend.java                Menu.java
AllDetails.dat               BoardingPassGenerator.java  PassengerCheck.java
AvailableFlights.dat         FlightDetails.java          Passengers.dat
AvailableFlights.java        Makefile                    UserFlight.dat
abhigyan@abhigyan-samsung:~/Code/Java/AirIndia$ 
```

For Java

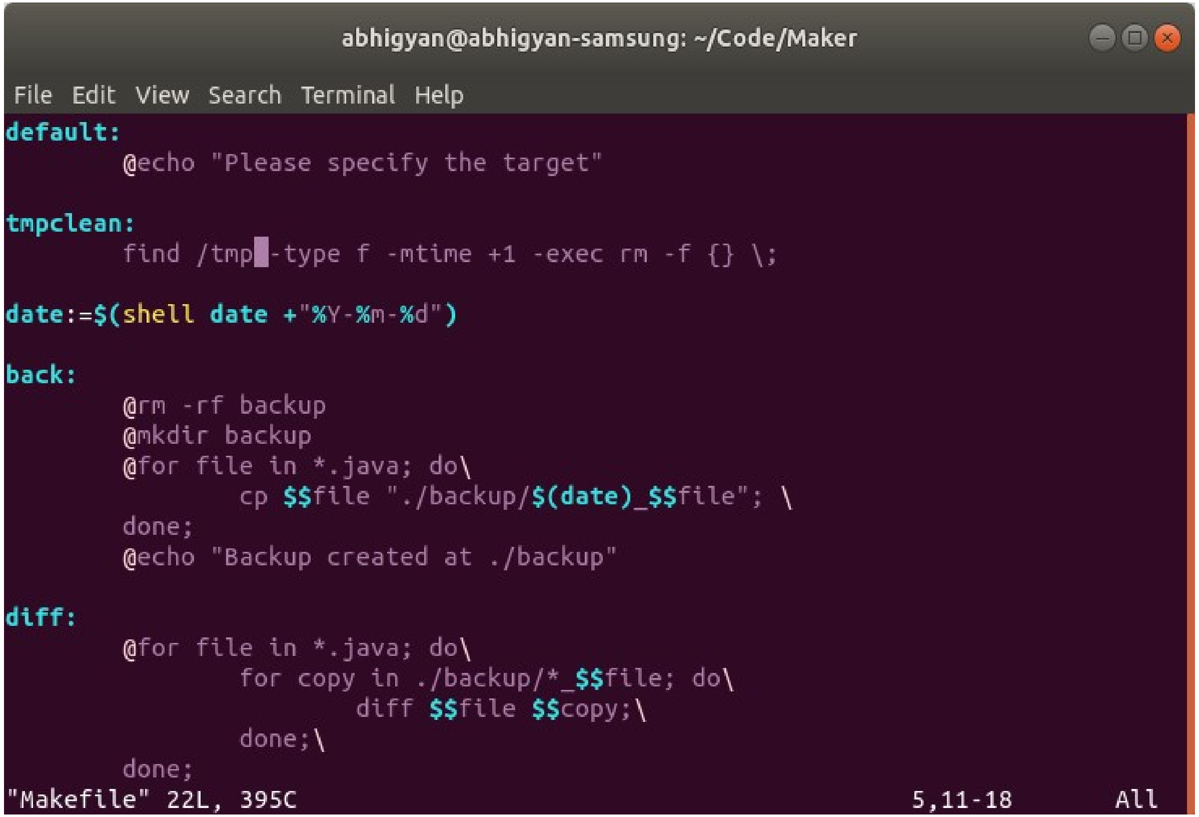
# 10.2. Create a makefile that uses a pattern for files rather than explicit listing of each of the files.

Now, we use the * wildcard to search through all the .h files and delete all the .out files in the current directory


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

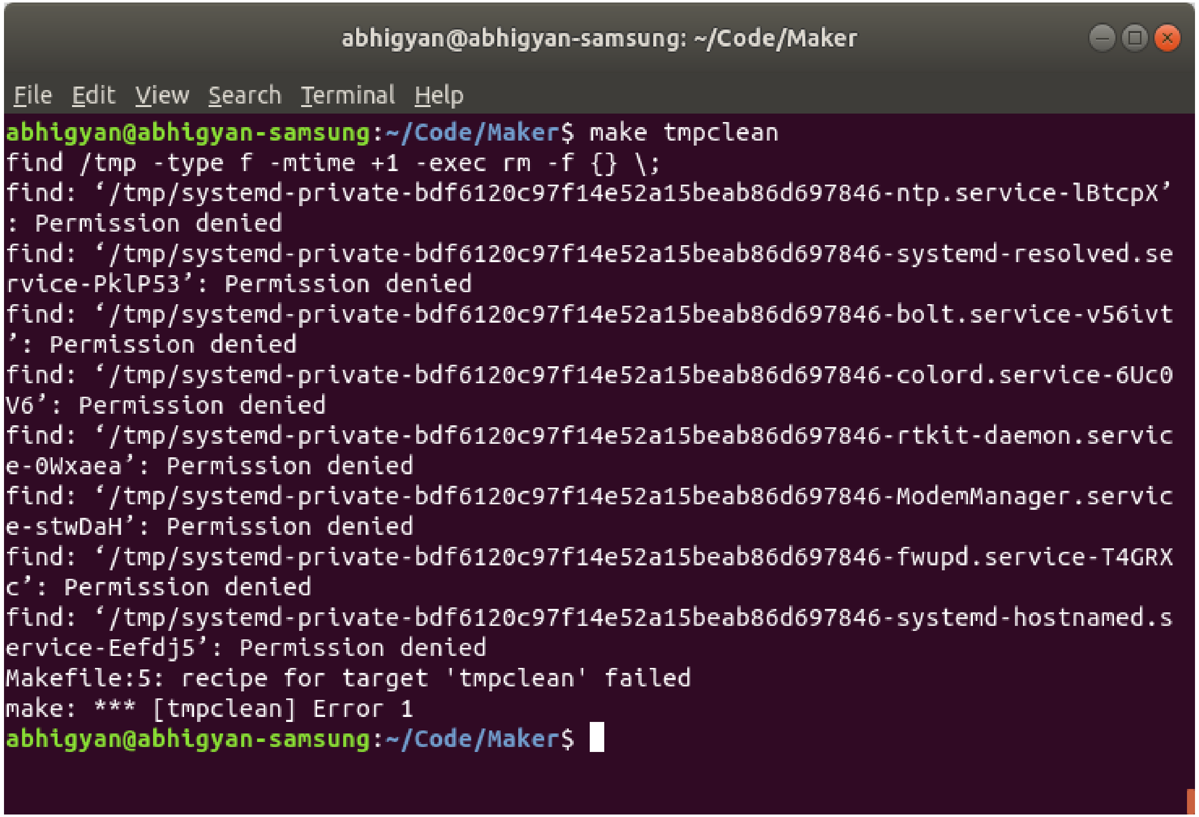
```
#Makefile

#shortcuts used:
cc = gcc #gcc compiler shortened to cc
cflags = -g3 -ggdb #this enables us to change all compiler options in one go

default:
        @echo "Please specify target first" #prevents user from making without s
pecifying target

#using the * wildcard to check for all files ending in .h and deleting all files
 ending in .out
math: *.h
        @rm -f *.out
        @$(cc) $(cflags) -o math.out mathematics.c

#in the following target, we touch a new file, put in some C code using the echo
 command, and then compile and show its output, all in a single make command!
cbrt: cbrtlib.h modlib.h
        @touch cuberooter.c;
        @echo "#include <stdio.h>\n#include \"modlib.h\"\n#include \"cbrtlib.h\"
\nint main(){\n\tprintf(\"%.010f\",cbroot(125));\n\treturn 0;\n}" > cuberooter.c
;
"Makefile" 27L, 1122C                                         1,1           Top
```

## 10.3. Create a Makefile that does simple book keeping tasks:


abhigyan@abhigyan-samsung: ~/Code/Maker
File Edit View Search Terminal Help

```
default:
        @echo "Please specify the target"

tmpclean:
        find /tmp -type f -mtime +1 -exec rm -f {} \;

date:=$(shell date +"%Y-%m-%d")

back:
        @rm -rf backup
        @mkdir backup
        @for file in *.java; do\
                cp $$file "./backup/$(date)_$$file"; \
        done;
        @echo "Backup created at ./backup"

diff:
        @for file in *.java; do\
                for copy in ./backup/*_$$file; do\
                        diff $$file $$copy;\
                done;\
        done;
"Makefile" 22L, 395C                                          5,11-18       All
```

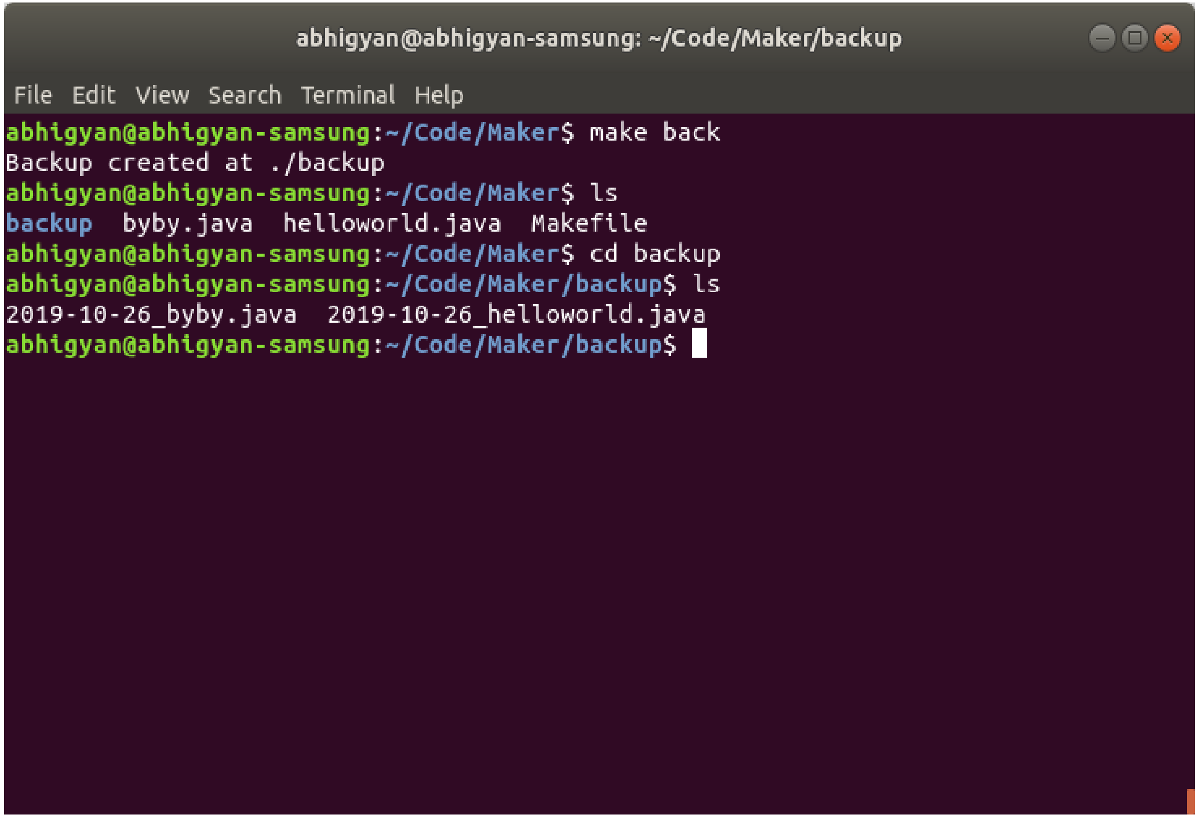
# 10.3.1 Create a Makefile that does simple book keeping tasks such as the following: Remove files older than a day from /tmp folder


abhigyan@abhigyan-samsung: ~/Code/Maker
File Edit View Search Terminal Help

```
abhigyan@abhigyan-samsung:~/Code/Maker$ make tmpclean
find /tmp -type f -mtime +1 -exec rm -f {} \;
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-ntp.service-lBtcpX’
: Permission denied
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-systemd-resolved.se
rvice-PklP53’: Permission denied
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-bolt.service-v56ivt
’: Permission denied
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-colord.service-6Uc0
V6’: Permission denied
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-rtkit-daemon.servic
e-0Wxaea’: Permission denied
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-ModemManager.servic
e-stwDaH’: Permission denied
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-fwupd.service-T4GRX
c’: Permission denied
find: ‘/tmp/systemd-private-bdf6120c97f14e52a15beab86d697846-systemd-hostnamed.s
ervice-Eefdj5’: Permission denied
Makefile:5: recipe for target 'tmpclean' failed
make: *** [tmpclean] Error 1
abhigyan@abhigyan-samsung:~/Code/Maker$
```

The command failed, as the sudo command wasn’t used, and thus left it as it was

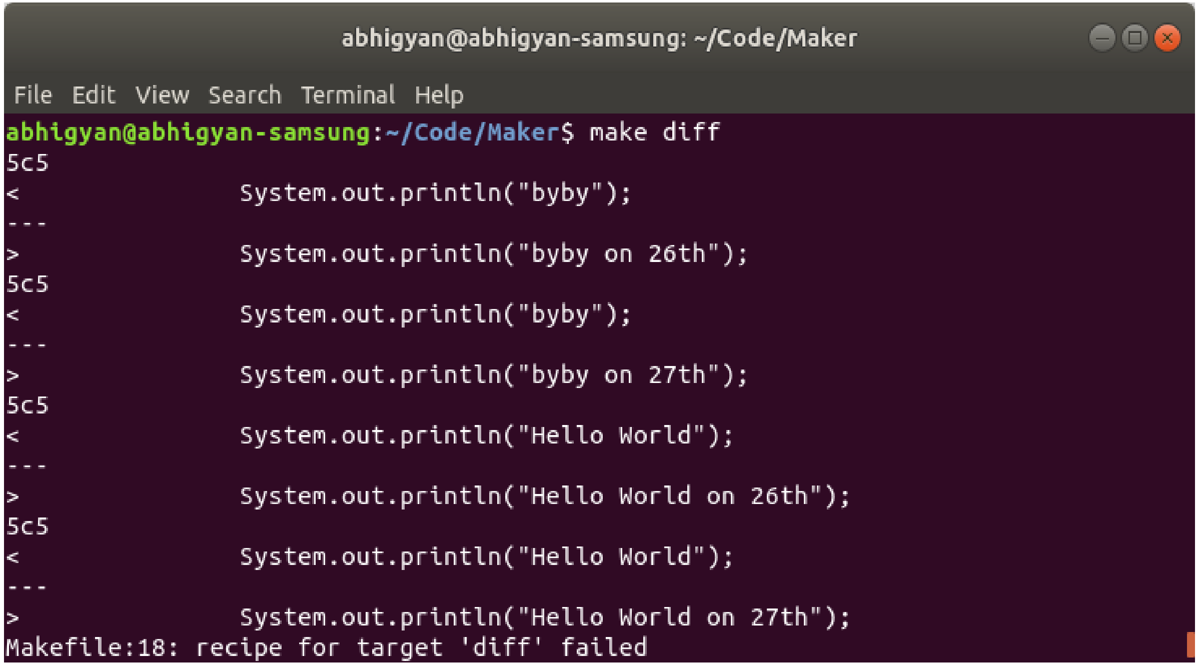
# 10.3.2 Create a Makefile that does simple book keeping tasks such as the following: Copy source codes to a backup folder using the date stamp in the filename itself.



abhigyan@abhigyan-samsung: ~/Code/Maker/backup
File Edit View Search Terminal Help

```
abhigyan@abhigyan-samsung:~/Code/Maker$ make back
Backup created at ./backup
abhigyan@abhigyan-samsung:~/Code/Maker$ ls
backup  byby.java  helloworld.java  Makefile
abhigyan@abhigyan-samsung:~/Code/Maker$ cd backup
abhigyan@abhigyan-samsung:~/Code/Maker/backup$ ls
2019-10-26_byby.java  2019-10-26_helloworld.java
abhigyan@abhigyan-samsung:~/Code/Maker/backup$ 
```

## 10.3.3 Create a Makefile that does simple book keeping tasks such as the following: Using the files in the backup folder, show the difference in the files that are modified recently.

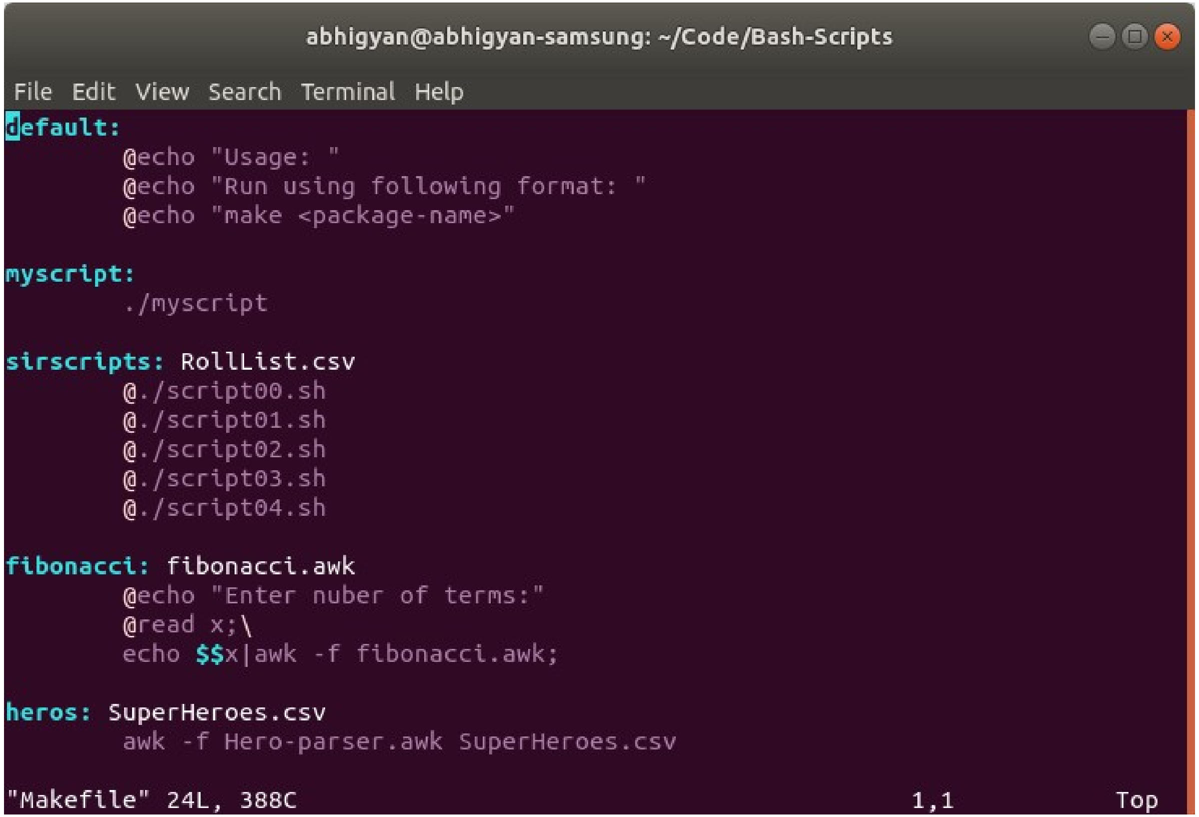


abhigyan@abhigyan-samsung: ~/Code/Maker
File Edit View Search Terminal Help

```
abhigyan@abhigyan-samsung:~/Code/Maker$ make diff
5c5
<               System.out.println("byby");
---
>               System.out.println("byby on 26th");
5c5
<               System.out.println("byby");
---
>               System.out.println("byby on 27th");
5c5
<               System.out.println("Hello World");
---
>               System.out.println("Hello World on 26th");
5c5
<               System.out.println("Hello World");
---
>               System.out.println("Hello World on 27th");
Makefile:18: recipe for target 'diff' failed
```

# Homework – Session 11

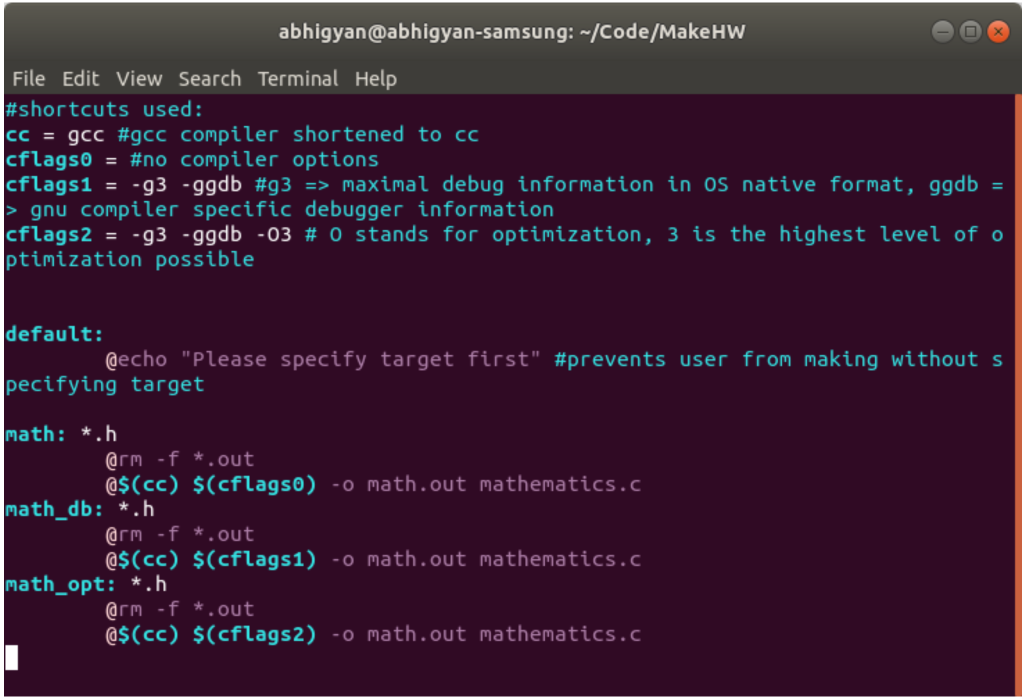
# 11.1. Create a Makefile that uses your own bash shell scripts in each recipe.



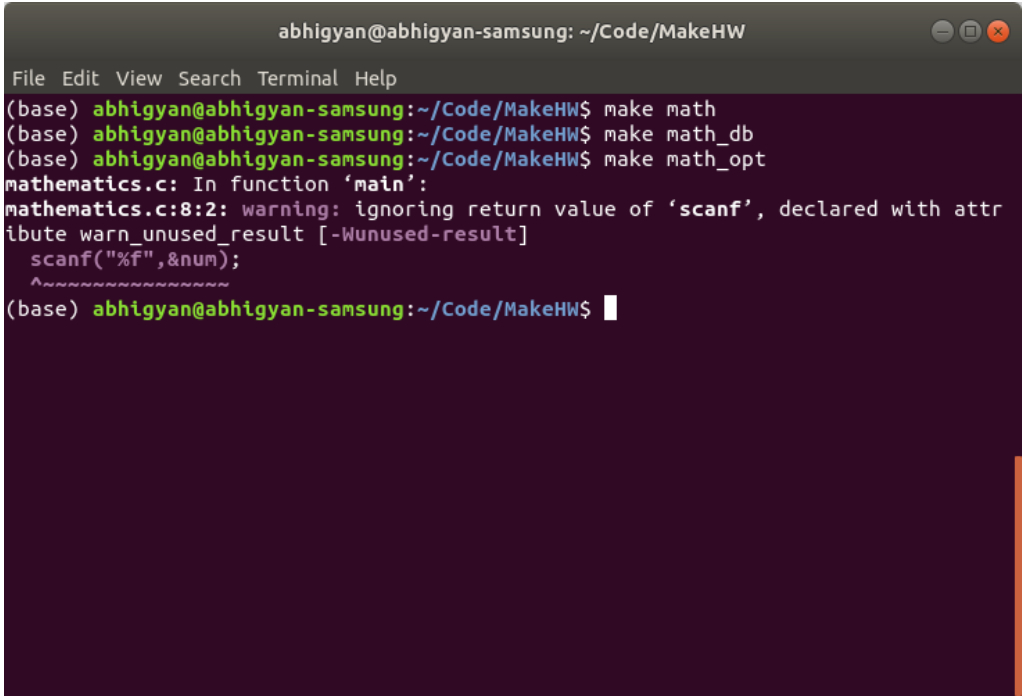
abhigyan@abhigyan-samsung: ~/Code/Bash-Scripts
File Edit View Search Terminal Help

```
default:
        @echo "Usage: "
        @echo "Run using following format: "
        @echo "make <package-name>"

myscript:
        ./myscript

sirscripts: RollList.csv
        @./script00.sh
        @./script01.sh
        @./script02.sh
        @./script03.sh
        @./script04.sh

fibonacci: fibonacci.awk
        @echo "Enter nuber of terms:"
        @read x;\
        echo $$x|awk -f fibonacci.awk;

heros: SuperHeroes.csv
        awk -f Hero-parser.awk SuperHeroes.csv

"Makefile" 24L, 388C                                          1,1           Top
```

# 11.2. Create a Makefile that can compile a code in three different ways namely (a) without any options, (b) with all options to provide warnings for non-compliance to ANSI standards etc., and (c) with options that help the final executable run fastest possible for the given architecture of the machine.


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

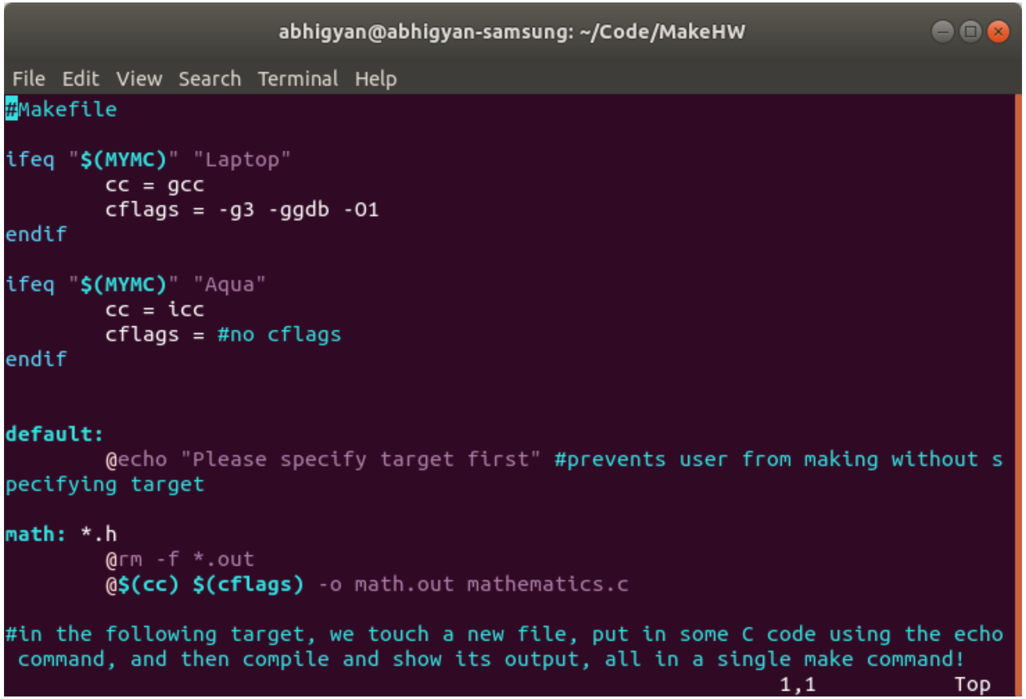
```
#shortcuts used:
cc = gcc #gcc compiler shortened to cc
cflags0 = #no compiler options
cflags1 = -g3 -ggdb #g3 => maximal debug information in OS native format, ggdb =
> gnu compiler specific debugger information
cflags2 = -g3 -ggdb -O3 # O stands for optimization, 3 is the highest level of o
ptimization possible


default:
        @echo "Please specify target first" #prevents user from making without s
pecifying target

math: *.h
        @rm -f *.out
        @$(cc) $(cflags0) -o math.out mathematics.c
math_db: *.h
        @rm -f *.out
        @$(cc) $(cflags1) -o math.out mathematics.c
math_opt: *.h
        @rm -f *.out
        @$(cc) $(cflags2) -o math.out mathematics.c
```


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

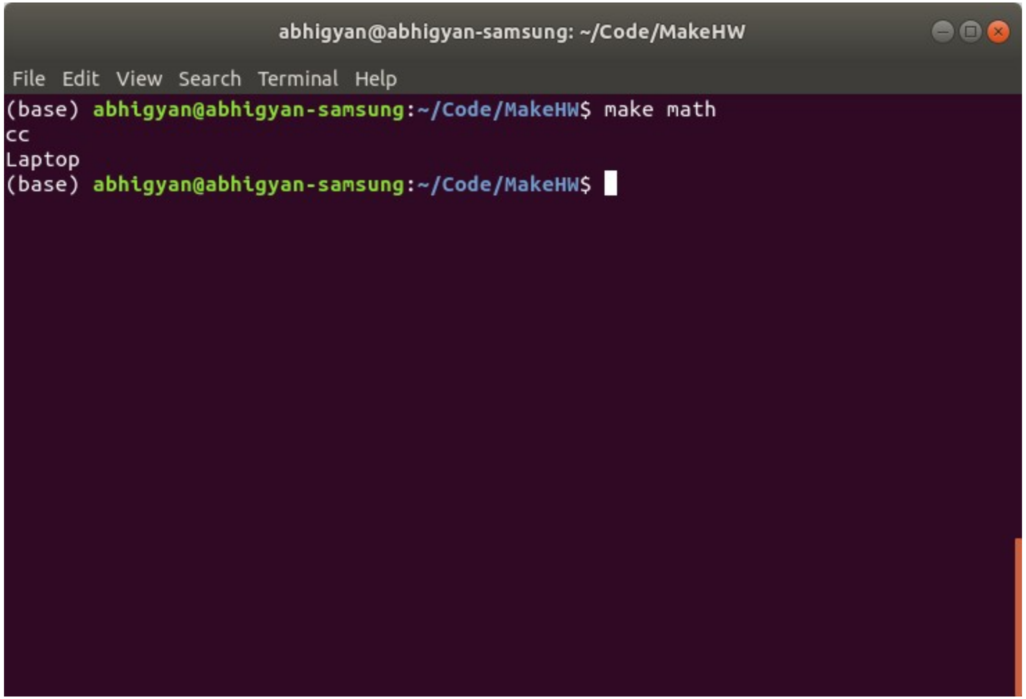
```
(base) abhigyan@abhigyan-samsung:~/Code/MakeHW$ make math
(base) abhigyan@abhigyan-samsung:~/Code/MakeHW$ make math_db
(base) abhigyan@abhigyan-samsung:~/Code/MakeHW$ make math_opt
mathematics.c: In function ‘main’:
mathematics.c:8:2: warning: ignoring return value of ‘scanf’, declared with attr
ibute warn_unused_result [-Wunused-result]
  scanf("%f",&num);
  ^~~~~~~~~~~~~~~~
(base) abhigyan@abhigyan-samsung:~/Code/MakeHW$
```

# Homework – Session 12

# 12.1. Prepare a Makefile that performs conditional compilation depending on the architecture of the machine. Ensure that this information is passed on to the compiler options explicitly. Use any of your old codes for this example.


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

```
#Makefile

ifeq "$(MYMC)" "Laptop"
        cc = gcc
        cflags = -g3 -ggdb -O1
endif

ifeq "$(MYMC)" "Aqua"
        cc = icc
        cflags = #no cflags
endif


default:
        @echo "Please specify target first" #prevents user from making without s
pecifying target

math: *.h
        @rm -f *.out
        @$(cc) $(cflags) -o math.out mathematics.c

#in the following target, we touch a new file, put in some C code using the echo
 command, and then compile and show its output, all in a single make command!
                                                                1,1           Top
```


abhigyan@abhigyan-samsung: ~/Code/MakeHW
File Edit View Search Terminal Help

```
(base) abhigyan@abhigyan-samsung:~/Code/MakeHW$ make math
cc
Laptop
(base) abhigyan@abhigyan-samsung:~/Code/MakeHW$
```