

Assignment 1: *datalab*

CSE4009: System Programming

Important: Virtualized x86 System

- This guide assumes that you have own virtualized x86 system
- This guide covers
 - Libraries required for this class
 - Hanyang VPN for **gitlab** access
 - Your **gitlab** account registration
 - Instructions for **datalab**
 - Submission guidelines

Libraries Required for This Assignment

- Required Libraries
 - git
 - gcc, gdb
 - build-essential
 - gcc-multilib, g++-multilib
- Check the libraries installed with the following script

```
$ sudo apt-get install -y git gcc gdb \  
build-essential gcc-multilib g++-multilib
```

Check Your Assignment

- Your **private** project has been prepared
 - 2025_CSE4009_{Student_ID}
 - Your git remote repository URL

2025_CSE4009_12769 / 2025_cse4009_2025012345

You can't push or pull repositories using SSH until you add an SSH key to your profile.

Add SSH key Don't show again

The Auto DevOps pipeline has been enabled and will be used if no alternative CI configuration is provided. Container registry is not enabled on this GitLab instance. Ask an administrator to enable it in the settings.

Settings More information

2 2025_cse4009_2025012345

1 Commit 1 Branch 0 Tags 3 KiB Project Storage

datalab assigned
Woong Sul authored just now

main 2025_cse4009_2025012345 / +

Auto DevOps enabled Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING

Add Kubernetes cluster Add Wiki Configure Integrations

Name	Last commit	Last update
01_datalab	datalab assigned	just now

http not git protocol!!

Check out Your Assignment

- Clone it to the project directory

- At your VM instance

```
$ mkdir -p Projects
```

```
$ cd Projects
```

```
$ git clone {your git remote repository}
```

```
$ cd {your directory}/01_datalab
```

```
wsul@splab:~/Projects$ git clone http://166.104.28.37/2025_cse4009_12769/2025_cse4009_2025012345.git
Cloning into '2025_cse4009_2025012345'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 18 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (18/18), 459.61 KiB | 1.59 MiB/s, done.
wsul@splab:~/Projects$ cd 2025_cse4009_2025012345/
wsul@splab:~/Projects/2025_cse4009_2025012345$ cd 01_datalab/
wsul@splab:~/Projects/2025_cse4009_2025012345/01_datalab$
```

Instructions for *datalab*

- You have 10 incomplete functions in **bits.c**
 - Complete the functions only with **! ~ & ^ | + << >>**

```
$ vim bits.c
```

```
137 //1
138 /*
139  * bitXor - x^y using only ~ and &
140  *   Example: bitXor(4, 5) = 1
141  *   Legal ops: ~ &
142  *   Max ops: 14
143  *   Rating: 1
144  */
145 int bitXor(int x, int y) {
146     return 2;
147 }
```

Instructions for *datalab* (Cnt'd)

- Prepare a binary executable using *make*

\$ make

```
[student20789@CSE4009-12843:~/Projects/2022_cse4009_201220789/datalab-handout]$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function 'test_function':
btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
  332 |         if (arg_test_range[1] < 1)
      |                     ~~~~~^~~~~
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
student20789@CSE4009-12843:~/Projects/2022_cse4009_201220789/datalab-handout$
```

\$./btest -f bitXor (@see the guide pdf file)

```
[student20789@CSE4009-12843:~/Projects/2022_cse4009_201220789/datalab-handout]$ ./btest -f bitXor
Score  Rating  Errors  Function
ERROR: Test bitXor(-2147483648[0x80000000],-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
Total points: 0/1
student20789@CSE4009-12843:~/Projects/2022_cse4009_201220789/datalab-handout$
```

\$./driver.pl (@self-test)

Provided with a detailed guide pdf file

Check the guide file

Submission Guideline

- Push only **bits.c** before the submission due

```
$ git add bits.c
$ git commit -m "your commit message.. here"
$ git push origin main
```

- Self-test example

```
wsul@splab:~/Projects/2025_cse4009_2025012345/01_data1ab$ ./driver.pl
1. Running './dlc -z' to identify coding rules violations.

2. Compiling and running './btest -g' to determine correctness score.
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c

3. Running './dlc -Z' to identify operator count violations.

4. Compiling and running './btest -g -r 2' to determine performance score.
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c

5. Running './dlc -e' to get operator count of each function.

Correctness Results      Perf Results
Points  Rating  Errors  Points  Ops    Puzzle
0       1       1       0       0     bitAnd
0       1       1       0       0     bitXor
0       1       1       0       0     isTmin
0       2       1       0       0     isEqual
0       2       1       0       0     negate
0       2       1       0       0     getByte
0       3       1       0       0     isLess
0       3       1       0       0     conditional
0       4       1       0       0     floatScale2
0       4       1       0       0     floatFloat2Int

Score = 0/30 [0/10 Corr + 0/20 Perf] (0 total operators)
```