

## Programming in C/C++

### Exercise Sheet 2

!!! Deadline: Tue, Nov 5<sup>th</sup>, 23:55 !!!

#### Task 01:

Given are the following source files:

```
/* scope1.c */
#include <stdio.h>

int globalvar = 1;

int main() {
    modtest();
    printf("globalvar = %d\n", globalvar);
}

/* scope2.c */
float globalvar;

void modtest(void) {
    globalvar = 2;
}
```

What output do you expect?

Save the two files separately and create a Makefile to compile and link these files to a single executable called “**scope**” and run it.

What’s the actual output? Why?

Look into the man page of the linker “ld”. Is there any method to prevent this or at least get notified? Try it, and report about it, but do not include the options in your Makefile.

Hint: gcc’s “-Wl,<option>” parameter is used to pass linker options through gcc to ld (spaces are replaced by “,”).

Submission: Submit your source code files and the Makefile, as well as your report

Points: (10pts)

### **Task 02:**

Read and understand the “right-left rule” [http://ieng9.ucsd.edu/~cs30x/rt\\_lt.rule.html](http://ieng9.ucsd.edu/~cs30x/rt_lt.rule.html)

What’s x in the following declarations? Apply the rules to everything in the declaration.

```
int (*x[])(int, int)
int *(*x(int*))()
int *(*(*x())[])(int)
int (*x)(int *(*)(int))
```

Submission: Put your report in your answer.txt file

Points: (report only, 20pts)

### **Task 03:**

Write a calculator program that accepts a single line input in form “<int> <op> <int>” with +, -, \*, / as <op> and executes this calculation with only integer types (division will round automatically). The program should run in a loop that is exited with a single ‘0’ in the input line above. See the sample session below for example of how your program should run.

Your program should be as robust as possible. Extra whitespaces in the input are ok, but do not accept any other additional characters not belonging to the arithmetic expression. So, don’t forget error checks.

Print "invalid input" when something is wrong with the input, but continue the program.

First try scanf() for the input and report the behaviour. Then change to a combination of fgets()/sscanf(). Look at the man pages how to use them. Only the latter solution has to be submitted.

Sample session:

```
Enter <int> <op> <int>, single '0' to exit: 1+2
1 + 2 = 3
Enter <int> <op> <int>, single '0' to exit: -5 - 3
-5 - 3 = -8
Enter <int> <op> <int>, single '0' to exit: 6*7
6 * 7 = 42
Enter <int> <op> <int>, single '0' to exit: 5/3
5 / 3 = 1
Enter <int> <op> <int>, single '0' to exit: x + y
invalid input
Enter <int> <op> <int>, single '0' to exit: 0
Goodbye
```

Submission: Name your executable **calc**

Points: (code 50 pts, report 10 pts, comments 10 pts)