C Programming basics

Lecture 4

Schedule

- Homework solving
- Functions
- Function arguments
- Global variables

Any questions ?

Homework solving

Questions ?

Functions

Functions

- Functions are used to encapsulate certain sequence of actions, that may provide a certain result based on certain arguments.
- Example : You want to calculate a squared of few numbers. The logic is the same for making the calculations, hence you make a function that is making that calculation and is providing a result.

Functions

result (function) type

name of function

arguments (variables passed to functions)

```
double sqare(double number){
    double result = number * number;
    return result;
}

int main(int argc, const char * argv[]) {
    double result = sqare(3.141592);
    printf("%lf\n", result);
    return 0;
}
```

Actions

We call the function and save the result to variable “result”

Functions

- Functions may be not returning a result. Sometimes when you need just to execute a sequence (like some check for correct inputed informations) you may not need to return a result, just to print a message to the user. The type of this function will be void.

```
void printMessage(){
    printf("Hello! Thanks for using the app. For making
it explode, please press 1 (;");
}
Then call : printMessage(); in the code.
```

Functions

- Demo :
  Create an application that is calculating quadratic equations.
  User inputs the coefficient before x^2, x^1 and X^0 and the program solves the equation.

Questions ?

Function arguments

Function arguments

- In order to make a function work, we really do need to be able to pass arguments to it. You can pass as many arguments as needed. In the most compilers that do support C and C++ standards you can make many functions with one name, but with different number of arguments, this is called overloading.

Function arguments (C++ compatible compiler)

```
int sum(int n1, int n2){
    return n1 + n2;
}

int sum(int n1, int n2, int n3){
    return n1 + n2 + n2;
}

int sum(int n1, int n2, int n3, int n4){
    return n1 + n2 + n3 + n4;
}
```

Function arguments

- The arguments in a function can be with different types. It is not required to have the same type for all arguments. The argument type can be different than the function type.

```
double myFunction(int arg1, char arg2, float arg3){
    return aDoubleNumber;
}
```

Questions ?

Global variables

Global variables

- Imagine that you need one variable to be changed or used in function A and the same variable to be changed or used in function B. How would you do that ?

Global variables

- Hence a declared variable is only valid in the function where it is declared you cannot use it in another function without passing it someway.
- For that reason if you want to use a variable in multiple functions without passing it as argument, you must declare it as global.

Global variables

- In order to declare a variable as global you must declare it outside of all functions. Best practice is to declare it just below #include and above the functions.

Questions ?

Global variables

- Task:
  Create 2 functions - void setup() and void loop(). setup() must execute only once upon starting the program. loop() must be executed after setup and must repeat indefinitely .
  The purpose of the program is to remember and print a character inputed by the user. This character is printed when the used inputs a new character, then the next character is replacing the old character.