ETH zürich

Session 05
Functions II

Informatik I

Lan Zhang

21.03.2025

Schedule

1. Review of Last Week
2. Main Topics
   - Functions, Libraries, Namespaces
   - Stepwise Refinement
3. In-Class Code Example
4. Past Exam Questions

Feedback

1. Loop analysis:
   - For: if predefined number of iterations is given
   - While: if no clear counting variable is given
   - Do-while: if loop body should be executed at least once regardless of condition
2. Approximation of Pi:
   - Avoid multiplication every iteration by going through the loop in intervals of 2
     - `0 < j < 2*m; j += 2 // j only takes on odd values`
   - Compute next term based on previous terms to avoid redundant calculations
   - The second approximation is more accurate as the series converges to pi more quickly. You can check the output for different m and compare the result against pi.
   - (In fact, the first approximation never reaches pi.)

[code]expert changes:

- Additional hidden testcases
- New input file, can be repeatedly used to run the program (with shortcut `f` in the terminal window)

Functions

A function is valid i.e. can be called after its first declaration.

Forward Declaration:

- Definition without `{ ... }`
- At the beginning of the program (before `main()`)
- e.g. `Type fname(type_1 pname_1,...);`

Separate Compilation

- Related Functions are declared and defined in external files in the working directory of the main file
- Declaration -> header file (*.h)
- Definition -> source file (*.cpp) -> object file (*.o)

Advantage: source code (.cpp) is no longer needed after the object code (.o) has been created

Functions from other files can be made available using `#include "filename.cpp/.h"`

Functions: Example

```
# include <iostream>
double f (double x) {
 return g(2.0 * x);
}
bool g (double x) {
 return x % 2.0 == 0;
}
void h () {
 std::cout << result;
}
int main () {
    double result = f(3.0);
    h();
    return 0;
}
```

Libraries and Namespaces

Libraries

- Grouping of similar functions
- Provide external data structures and algorithms

Namespaces

- Used to categorize functions, types, etc. (e.g. for different usages)
- Avoid conflicts with identical function names

`-> see week 5 code example <CPP Headers & Namespaces>`

Standard Library

- Functions and data structures supplied by the compiler
  - Functions `-> std::sin(x), std::sqrt(x)`
  - Objects `-> std::vector<T>, std::array<T, N>`
  - Algorithms `-> std::sort(...)`

Stepwise Refinement

- Divide the problem into sub-problems
- Start at a high level of abstraction (comments and pseudocode)
- Implement functionalities step by step

Exit codes in C++:

- Exit status, integer value returned by a program upon termination
- Used by the program's caller or user to determine whether the program's execution was successful

0	-	successful program termination
-6	-	assert(...) failed
-8	-	division by 0
-9	-	program was killed due to memory overflow
-11	-	segmentation fault

In-Class Code Example

[Exam 2022.02 (MAVT + ITET)] Decimal to arbitrary base

Convert a number n >= 0 in base 10 to its representation in base b with 1 < b < 10.

```
int convert_base(int n, int b) {
     int res = 0;
     int pos = 1;
     while (n != 0){
           res += pos * (n % b); //  next digit in base b
           n /= b; // remove last digit
           pos *= 10; // progressing from lowest to highest digit
     }
     return res;
}
```

`#pragma once`: preprocessor directive in C and C++ that ensures a header file is included only once in a single compilation

Past Exam Questions

[Exam 2022-08 MAVT] Expression Evaluation

Remark to Type and Value Questions:

The keyword auto means that the type of the expression is determined by the compiler. In the following it thus stands for the expression type that you need to identify.

Provide type and value of variable c.

```
int a = 5;
int b = 1;
auto c = (9 * a + b) % a;
```

```
int a = 5;
double b = 1;
auto c = (9.0 * a + b) / a;
```

Past Exam Questions

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system

F*($\beta = 2, p = 3, e_{min} = -1, e_{max} = 4$)

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- `1.25` can be represented exactly in the floating point system F*
- There is no number Z in F such that `0.0625 < Z < 0.25`
- `3.25` can be represented exactly in the floating point system F*

Past Exam Questions

[Exam 2022-08252-08(47/48/56)] Loop Termination

```
int sum = 17;
int i = 1;

do {
i += sum;
sum =  sum / 2;
} while (i > sum && sum >= 0) ;
std: : cout << sum;
```

Which statement describes the output best?

- 17
- 8
- Never terminates
- 18