

Assignment 2 - More Overloading, Strings and Classes

- The problems of this assignment must be solved in C++.
- The TAs are grading solutions to the problems according to the following criteria:
<https://grader.eecs.jacobs-university.de/courses/320142/2018.2r2/Grading-Criteria-C++.pdf>

Problem 2.1 *Function overloading*

(1 point)

Presence assignment, due by 18:30 h today

Write three overloaded functions ... `myfirst(...)` which should do the following:

- 1) if called with an array of integers, it determines and returns the first positive and odd value from the array. If no such element exists then -1 should be returned;
- 2) if called with an array of doubles, it determines and returns the first positive element which does not have a fractional part. If no such element exists then -1.1 should be returned;
- 3) if called with an array of chars, it determines and returns the first element which is a vowel. If no vowels are present in the array then the character '0' should be returned.

Write a program which calls the above functions and illustrates their effect. You may choose to enter test data from the keyboard or to initialize variables within your code.

You can safely assume that the input will be valid.

Problem 2.2 *Swapping with call-by-reference*

(1 point)

Presence assignment, due by 18:30 h today

Write a program `swapping_fct.cpp`, which provides three overloaded functions `swapping(...)`.

These functions should swap two ints, two doubles, and two pointers to char. The swapping should be done by a "real" call-by-reference (i.e., not by using the * type). Complete the following code fragment:

```
#include <iostream>

using namespace std;

void swapping(...) { .... } // swap ints
void swapping(...) { .... } // swap doubles
void swapping(...) { .... } // swap char pointers

int main(void) {
    int    a = 5,    b = 10;
    double x = 5.2, y = 10.7;
    const char *str1 = "One";
    const char *str2 = "Two";

    cout << "a=" << a << ", b=" << b << endl;
    cout << "x=" << x << ", y=" << y << endl;
    cout << "str1=" << str1 << ", str2=" << str2 << endl;

    swapping(a, b);
    swapping(x, y);
    swapping(str1, str2);
}
```

```

cout << "a=" << a << ", b=" << b << endl;
cout << "x=" << x << ", y=" << y << endl;
cout << "str1=" << str1 << ", str2=" << str2 << endl;
return 0;
}

```

Problem 2.3 *Dynamic allocation and references*

(1 point)

Write a program which reads from the keyboard an integer n followed by n integer values which are to be stored in a dynamically allocated array x . Your program should define a function `void add_min(...)` for determining the minimum value in the array and adding this minimum to all elements of the array. Write another function `void deallocate(...)` for deallocating the memory occupied by an array. The `main()` function should allocate memory for the array, call the first function, **illustrate its effect by printing the values of the array** and finally deallocate the memory occupied by the array by calling the second function. *You can safely assume that the input will be valid.*

Problem 2.4 *Word guessing*

(2 points)

Write a program that stores a list of words ("computer", "television", "keyboard", "magazine", "book") and 10 other words of your choice in an array of strings. Inside of a game loop your program should **randomly** choose one word out of the 15 possible words. The program should print the word on the screen after replacing all vowels by underscores, then the player should try to guess the word. After the player has guessed the right word, the number of tries should be printed on the screen and the player should be asked whether he/she wishes to play again. If the player enters "quit" as a word guess, then the game should immediately stop. *You can safely assume that the input will be valid and that "quit" will not be in the set of words to be guessed.*

Problem 2.5 *Checking words for palindrome*

(1 point)

A palindrome is a word that is the **same read** forward and backwards (e.g., "level", "ana", "civic", etc.). Write a function `bool isPalindrome(string s)` that recognizes this characteristic of a string by returning `true` or `false`. Your program should have the form of a game as the previous program with the difference that the player enters a word and then the function should be called to check if word is a palindrome or not. This should be repeated until "stop" is entered as a word (i.e., then the game should immediately stop). *You can safely assume that the input will be valid.*

Problem 2.6 *Error messages provided by the compiler*

(1 point)

Download and use the files:

<https://grader.eecs.jacobs-university.de/courses/320142/cpp/Critter.h>

<https://grader.eecs.jacobs-university.de/courses/320142/cpp/Critter.cpp> and

<https://grader.eecs.jacobs-university.de/courses/320142/cpp/testcritter.cpp>.

You will need to create a project to successfully compile all three files (or compile them from the command line as specified on the slides).

- Comment out the `using namespace std;` and then take your time, read and interpret the error messages.
- Also remove the `Critter::` prefix in one of the methods in `Critter.cpp`, read and interpret the error message.

Then create a file named `explanations.txt`. This file should be uploaded together with the other files and should contain your descriptions and interpretations of the errors as well as your comments on potential alternative solutions.

You can assume that the input will be valid.

Problem 2.7 *The Critter class*

(1 point)

Use the previously given files: `Critter.h`, `Critter.cpp` and `testcritter.cpp`.

Expand `Critter.h` by two additional properties of your choice, and corresponding setter and getter methods, then adjust `Critter.cpp` and `testcritter.cpp` accordingly.

Also adapt the `print()` method such that the new properties are printed on the screen as well. You can safely assume that the input will be valid.

How to submit your solutions

- Your source code should be properly indented and compile with `g++` without any warnings (You can use `g++ -Wall -o program program.cpp`). Insert suitable comments (not on every line ...) to explain what your program does.
- Please name the programs according to the suggested filenames (they should match the description of the problem) in Grader.
Each program **must** include a comment on the top like the following:

```
/*  
    CH08-320142  
    a2.p1.cpp  
    Firstname Lastname  
    myemail@jacobs-university.de  
*/
```

- You have to submit your solutions via Grader at **`https://grader.eecs.jacobs-university.de`**.
If there are problems (but **only** then) you can submit the programs by sending mail to `k.lipskoch@jacobs-university.de` **with a subject line that begins with CH08-320142**.
It is important that you do begin your subject with the coursenummer, otherwise I might have problems to identify your submission.
- Please note, that after the deadline it will not be possible to submit any solutions. It is useless to send late solutions by mail, because they will not be accepted.

This assignment is due by Wednesday, November 21st, 10:00 h.