

Tutorial 5a

Katja Mankinen

October 5, 2017

Aim

After this tutorial, you

- can branch your code based on different conditions
- know how to use `for` and `while` loops
- can store data in vectors
- have gathered much knowledge for your final project!

You should preferably finish all the exercises today. It may look like a lot of work, but it is actually very doable! **You can start doing the exercises immediately, at your own pace.** If you have some problems or questions, just ask help! A summary of exercises will be updated to Live@Lund with example solutions after the tutorial.

1. Warm up with loops

As a warmup exercise, let's do a short exercise on different loops. Create a new code file, include the needed libraries, add a main function, add the following loops, compile the program (`g++ -o warmup warmup.cpp`), and check that the output is as expected.

1. Using `for`, print the following shape:

```
*
**
***
****
*****
```

2. Add a `while` loop that will ask numbers from an user until the number is less than 10.
3. Extra: modify your `while` loop so that instead of numbers, it asks your favourite animals, and ends when user writes "cat".

2. Dates

Write a new C++ program that accepts a month and a day as user input, and displays the following two prompts:

```
Enter a month:
Enter a day of the month:
```

Program should accept and store a number in the variable `month` in response to the first prompt, and accept and store a number in the variable `day` in response to the second prompt. Use 1 for January, 2 for February and so on. If the month entered is not between 1 and 12, print a message informing that an invalid month has been entered. If the day is not between 1 and 31, print a message informing that an invalid day has been entered.

Compile and run your code as you have learnt in previous lectures and tutorials:

```
g++ -o dates dates.cxx
./dates
```

Make sure that all program printouts make sense and are correct in your opinion. Then answer the following questions:

- What happens if the user enters a number with a decimal point for the month?
- How can you make sure your `if` statements check for an integer number?
- How would you modify your code if you want to enter many different dates after each other and check their validity? (no need to make actual modifications, but think about it!)

3. Leap years

Years that are evenly divisible by 400 or are evenly divisible by 4 but not by 100 are called leap years, such as 2016 and 1860. In leap years, February has 29 days and the other months have either 30 or 31 days.

Write a C++ program that accepts the year as user input, determines whether the year is a leap year or a common year, and displays a message telling whether the entered year is a leap year or not. Users can enter as many years as they want until they type some defined number or string (define yourself!). Compile and run your code, and make sure the output is as expected.

```
g++ -o leapyears leapyears.cxx
./leapyears
```

Example output:

```
Enter a year: 1961
It is not a leap year!

Enter a year: 2020
It is a leap year!

Enter a year: bye
Ok, you want to quit this amazing leap year program. Bye!
```

Hints!

- Are you stuck with the algorithm? Just simple `if` clauses will do. Still stuck? Use the magic power of Google and Wikipedia.
- You can test your algorithm with the following leap years: 2356, 1896, 2028, 2204, 1812, ...

4. Vectors and strings with famous scientists

Your next exercise is to make a list of famous scientists. Name of each scientist will be represented by a string stored in a container.

<i>Birthday</i>	<i>Name</i>	<i>Occupation</i>
1815-12-10	Ada Lovelace	mathematician and world's first computer programmer
1878-11-07	Lisa Meitner	physicist, worked on radioactivity and nuclear physics
1912-05-31	Chien-Shiung Wu	physicist, worked on nuclear physics
1914-11-09	Hedy Lamarr	actress and inventor, a pioneer in the field of wireless communications
1928-07-23	Vera Rubin	astronomer, pioneered work on galaxy rotation curves
1941-06-07	Cecilia Jarlskog	theoretical physicist, working mainly on elementary particle physics
1943-07-15	Jocelyn Bell Burnell	astrophysicist, discovered the first radio pulsars
1960-10-29	Fabiola Gianotti	particle physicist, CERN Director-General

Create a C++ program that does the following:

1. Create an empty vector of type `string`. Print it.
2. Append the first four names (Lovelace, Meitner, Wu, Lamarr) to the list. Print it.
3. Remove the last name from the vector. Print it.
4. Append the name "Rubin" to the vector. Print it.
5. Write a loop that searches for the name "Meitner" and replace it with "Jarlskog". Print the vector.
6. Append the names "Bell Burnell" and "Gianotti" to the vector. Print it.
7. Write a loop that searches for the name "Wu" and then remove it from the vector. (You are permitted to reorder the vector to accomplish the removal, if you want.) Print the vector.

Extra exercise for those who finish fast!

- Make a text file containing the names, one name per line. Modify your program so that it reads the names from the text file and appends them to the vector.
- There are many ways to do this!
- One is to use `std::getline()` to read the entire line into a `std::string`, and then possibly construct a `std::stringstream`. (what is the pitfall here?)
- Another is to use `cin`. (what is the pitfall here?)
- Don't be afraid of googling: Stack Overflow is your friend. Just remember: do try to understand what is written there instead of just naive copy-pasting.
- Is any of the mentioned famous scientists born in a leap year?

5. Warm up with pointers

Write a following program, compile it, and make sure you understand all the outputs!

```
#include <iostream>

using namespace std;

int main(){
    cout << "This is your very first pointer program!" << endl;
```

```

int a = 5, b = 10;
int *p1;
int *p2;
p1 = &a;
cout << p1 << " = " << &a <<endl;
p2 = &b;
cout << p2 << " = " << &b<<endl;
*p1 = 10;
cout << *p1 << " = " << a << endl;
p1 = p2;
cout << p1 << " = " << p2 <<endl;
*p1 = 20;
cout << "*p1" << *p1 << "\np1" << p1;
cout << "\np2" << p2 << "\n*p2 " << *p2;
cout << "\na " << a << "\nb " << b;
cout << "\n&a " << &a << "\n&b " << &b << endl;
}

```

6. Passing by value and reference

Your last task is to finish my incomplete program. Some of its functionalities are not working yet. Copy the code below, and add the missing parts!

```

#include <iostream>

void calculateBirthYear(int myAge){
    int birthYear = 2017 - myAge;
    std::cout << "birthYear: " << birthYear << std::endl;}

void increaseAgeByRef(int &myAge){
    //I want to increase my age by 1}

void increaseAgeByPtr(int *myAge){
    //I want to increase my age by 1}

void increaseAge(int myAge){
    //I want to increase my age by 1}

int main()
{
    int myAge = 30;
    std::cout << myAge << std::endl;
    calculateBirthYear(myAge);
    increaseAge(myAge); // Call increaseAge() to increase my age by 1
    std::cout << myAge << std::endl; //is this working?
    //call different functions here and print the output; finally I want to be 31
    return 0;
}

```