PROGRAMMING LOGIC AND TECHNIQUES

Tools, programming work, and ethics

Tools for programmers

- Integrated development environments (IDE)
  - Eclipse
  - Visual Studio
- Revision control systems: servers and clients
  - Subversion (SVN)
  - Tortoise
  - Git

Tools for programmers

- Database management systems: servers and clients
  - MySQL
  - MySQL Workbench
  - PostgreSQL
  - MSSQL

Tools for programmers

- Project management tools
  - Trello
  - Microsoft Project
  - Jira
  - Redmine

Tools for programmers

- Graphics editors
  - Gimp
  - Photoshop
  - Illustrator

Teamwork

- Important rules to follow when working in a team:
  - Respect your teammates
  - Fulfill your responsibilities and complete your tasks
  - Communicate well
  - Focus first and foremost on the result
  - Do not try to prove that you are right at all costs
  - Understand that the role of participant is as essential and important as the role of leader
  - Do your part

Teamwork in school

- When working in teams for class, it is important that everyone apply themselves fully and that the teams maintain a level of **constant communication**.
- It often happens that certain members of a team isolate themselves and don't talk to anyone before the end of the project, and remain caught up in problems they don't manage to resolve.
- No one is here to judge what you're able or unable to do. If a task appears too complex and you are stuck, you need to **talk about it** with the members of your team. Problems seen from another angle by someone else can very often lead quickly to an unexpected solution.
- **Communication is key**!

Teamwork in school

- Sometimes you will be able to choose the people you team up with, sometimes the teams will be formed randomly, and sometimes the teacher will be responsible for forming the teams.
- When planning group work, you must learn to subdivide the work into multiple subtasks. This is never easy in the beginning, but the earlier you begin clearly defining who will do what in your teams, the faster you will improve your project management skills.

Teamwork in school

- It's certain that sometimes the division of labor won't end up being perfectly equal, due to not having enough knowledge and experience to properly evaluate how much time should be allocated to a specific task you've never done before.
- But it is best to assign the task immediately, and then, if you later reevaluate it as being significantly more complex/simple than it originally appeared to be, be sure to discuss it again with your teammates as soon as possible!
- https://ed.ted.com/lessons/how-to-manage-your-time-more-effectively-according-to-machines-brian-christian

Teamwork in the workplace

- The people currently around you will not only be your classmates for the duration of your studies, but are also future industry contacts who will know what you are capable of accomplishing.
- You're not only here to get a passing grade for the course – you're here to **learn a profession** and develop a solid network of contacts (students, teachers, etc…), who will help you progress in your career.
- If you are not engaged in your teamwork while in school, what certainty will we have that you will apply yourself once you reach the job market?

Teamwork in the workplace

- In the job market, it is rare for you to have the opportunity to choose the team you work with.
- The concept of a team also varies greatly from one company to another:
  - Freelance: 1 programmer who works in a team with the client, and possibly with other freelancers, or sometimes with the technical team already in place at the client company
  - Small company: 1 to 3 programmers in a diverse team
  - Medium-sized company: 1 to a few teams of programmers
  - Large company: Several teams of programmers

Teamwork in the workplace

- Furthermore, it is important to remember that a programmer or a team of programmers will not only be working with other programmers! They will also have to work with other teams possessing various skills, e.g., marketing, accounting, education, etc.
- For example, it is not unusual for a company that develops financial products to have teams in place with financial competencies that can share their knowledge and experience, in order to obtain a product that is as well adapted to the industry as possible.

Work progress

- Here are two very important points to consider while work is in progress:

- **Respecting deadlines:**
  - Depending on the project, there are often several steps to take before completion. Of course, the final delivery of the project is important, but sometimes certain specific tasks need to be completed before you can pass on to the following ones.
  - It is important to identify these tasks early on to ensure that they do not block the entire project's progress.

Work progress

- **Respecting functionalities:**
  - When the analysis of the project is performed, the analyst and the client establish a list of functionalities, with the goal of obtaining an application that is complete and that responds to the needs of the client.
  - These functionalities are similar to clauses in a contract that must be respected.
  - When all of the functionalities have been developed in accordance with the client's needs, the client will test the application, approve it, and pay the invoice (according the terms of the agreement). Only then will the application be deployed or installed.

Work progress

- A common mistake programmers make when developing an application:
  - You conceive of a functionality that could be very useful, but that hasn't necessarily been approved by the client.
  - Regardless, you implement the feature, thinking to yourself, "It's great, the client will love it!".
  - This can go in many different ways:

Work progress

1. The customer sees the functionality, loves it and everything goes well.
2. The customer sees the functionality and does not understand it.

   The analyst doesn't understand it either, and the analyst must then speak with the head developer, who will consult you to understand what you wanted to accomplish. After the explanation, the analyst and the client are updated. The client may or may not accept the functionality – and if not, then many people have lost time for nothing!
3. The customer sees the functionality, loves it and everything goes well… but the functionality causes bugs, unforeseen delays, failure to meet deadlines.

   It unexpectedly requires the configuration of the final deployment/installation environment to be restructured, and even causes project cost overruns. The project manager is instructed to remove the functionality from the application. The client then looks for it but can't find it even though they *loved* it, and then blames the analyst for not thinking of it in the first place.

Work progress

- In conclusion, if, as a programmer, you see that a useful functionality could be added to the application, remember that **it is not up to you to decide to implement it**.

- If anything, this functionality may have already been refused by the client, or maybe the project management team established that this task had financial or technical implications that the client could not afford to accept, either now or ever.

Confidentiality of information

- In software development, you will often be exposed to privileged or confidential information that belongs to your employers or to their clients.
- Under no circumstances should you use this information for any purpose other than the development requested, and in no case should you use this information for any personal purpose whatsoever.
- These are legal considerations, and violating these principles **can lead to criminal charges**. So be prudent.

Respecting privacy

- Specifically, medical records, educational records and financial records have stringent laws about who is allowed to access them.

- Respecting anonymous user data is also important. Doxing and data re-identification are frowned upon at best, illegal at worst.

Intellectual property of developed systems

- In general, when you are employed and paid to develop a computer system, the system you develop belongs to your employer.
- The source code, the analysis documents, the compiled code, as well as any other things that you produce within the scope of your employment – all of these **do not belong to you**. You cannot use them for personal purposes or leave with them when you change jobs.

Hacking

- It is illegal.
- There is **no** valid reason to attempt an intrusion into a network or onto a server without authorization. Not only will you lose your job, but in the vast majority of cases, you will be sued.
- Hacker attacks that were done “just to see”, “simply to test”, or that were “harmless”, have ended many careers in the field.

Hacking

- The myth that the CIA or the FBI will offer you a position as a security expert if you succeed in breaking into a system is just that…a myth.
- Instead, you will end up in prison with an injunction from the court that forbids you from using a computer. Good luck to your career after that.
- This also applies to the school. The rules are not different just because you are students.
- **Hacking will not be tolerated.**