



C Piscine

Shell 00

*Summary: This document contains the subject matter for the Shell module 00 of the C Piscine at 42.*

*Version: 5.0*

# Contents

I	Instructions	2
II	AI Instructions	3
III	Foreword	5
IV	Exercise 00: Z	7
V	Exercise 01: testShell00	8
VI	Exercise 02: Oh yeah, mooore...	9
VII	Exercise 03: SSH me!	11
VIII	Exercise 04: midLS	12
IX	Exercise 05: GiT commit	13
X	Exercise 06: gitignore	15
XI	Exercise 07: diff	16
XII	Exercise 08: clean	17
XIII	Exercise 09: Illusions, not tricks, Michael...	18
XIV	Submission and peer-evaluation	19

# Chapter I

## Instructions

- These exercises are carefully arranged in order of difficulty, from easiest to hardest. We will not consider a successfully completed harder exercise if an easier one is not perfectly functional.
- Ensure that you have the appropriate permissions on your files and directories.
- You must follow the **submission procedures** for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- Additionally, your exercises will be checked and graded by a program called **Moulinette**.
- **Moulinette** is extremely meticulous and strict in its evaluation. It is entirely automated, and there is no way to negotiate with it. To avoid unpleasant surprises, be as thorough as possible.
- Shell exercises must be executable with `/bin/sh`.
- You must not leave any additional files in your directory other than those specified in the assignment.
- Have a question? Ask the peer on your right. If not, try the peer on your left.
- Your reference guide is called **Google / man / the Internet / ...**
- Examine the examples carefully. They may contain details that are not explicitly mentioned in the assignment.

# Chapter II

## AI Instructions

### ● Context

The C Piscine is intense. It's your first big challenge at 42 — a deep dive into problem-solving, autonomy, and community.

During this phase, your main objective is to build your foundation — through struggle, repetition, and especially **peer-learning** exchange.

In the AI era, shortcuts are easy to find. However, it's important to consider whether your AI usage is truly helping you grow — or simply getting in the way of developing real skills.

The Piscine is also a human experience — and for now, nothing can replace that. Not even AI.

For a more complete overview of our stance on AI — as a learning tool, as part of the ICT curriculum, and as a growing expectation in the job market — please refer to the dedicated FAQ available on the intranet.

### ● Main message

- 👉 Build strong foundations without shortcuts.
- 👉 Really develop tech & power skills.
- 👉 Experience real peer-learning, start learning how to learn and solve new problems.
- 👉 The learning journey is more important than the result.
- 👉 Learn about the risks associated with AI, and develop effective control practices and countermeasures to avoid common pitfalls.

## ● Learner rules:

- You should apply reasoning to your assigned tasks, especially before turning to AI.
- You should not ask for direct answers to the AI.
- You should learn about 42 global approach on AI.

## ● Phase outcomes:

Within this foundational phase, you will get the following outcomes:

- Get proper tech and coding foundations.
- Know why and how AI can be dangerous during this phase.

## ● Comments and example:

- Yes, we know AI exists — and yes, it can solve your projects. But you're here to learn, not to prove that AI has learned. Don't waste your time (or ours) just to demonstrate that AI can solve the given problem.
- Learning at 42 isn't about knowing the answer — it's about developing the ability to find one. AI gives you the answer directly, but that prevents you from building your own reasoning. And reasoning takes time, effort, and involves failure. The path to success is not supposed to be easy.
- Keep in mind that during exams, AI is not available — no internet, no smartphones, etc. You'll quickly realise if you've relied too heavily on AI in your learning process.
- Peer learning exposes you to different ideas and approaches, improving your interpersonal skills and your ability to think divergently. That's far more valuable than just chatting with a bot. So don't be shy — talk, ask questions, and learn together!
- Yes, AI will be part of the curriculum — both as a learning tool and as a topic in itself. You'll even have the chance to build your own AI software. In order to learn more about our crescendo approach you'll go through in the documentation available on the intranet.

### ✓ Good practice:

I'm stuck on a new concept. I ask someone nearby how they approached it. We talk for 10 minutes — and suddenly it clicks. I get it.

### ✗ Bad practice:

I secretly use AI, copy some code that looks right. During peer evaluation, I can't explain anything. I fail. During the exam — no AI — I'm stuck again. I fail.

# Chapter III

## Foreword

Below are the lyrics to *City Hunter*'s theme song, "*Moonlight Shadow*":

The last time ever she saw him  
Carried away by a moonlight shadow  
He passed on worried and warning  
Carried away by a moonlight shadow.  
Lost in a riddle that Saturday night  
Far away on the other side.  
He was caught in the middle of a desperate fight  
And she couldn't find how to push through

The trees that whisper in the evening  
Carried away by a moonlight shadow  
Sing a song of sorrow and grieving  
Carried away by a moonlight shadow  
All she saw was a silhouette of a gun  
Far away on the other side.  
He was shot six times by a man on the run  
And she couldn't find how to push through

[Chorus]

I stay, I pray  
See you in Heaven far away...  
I stay, I pray  
See you in Heaven one day.

Four A.M. in the morning  
Carried away by a moonlight shadow  
I watched your vision forming  
Carried away by a moonlight shadow  
A star was glowing in the silvery night  
Far away on the other side  
Will you come to talk to me this night  
But she couldn't find how to push through

[Chorus]

Far away on the other side.  
Caught in the middle of a hundred and five  
The night was heavy and the air was alive  
But she couldn't find how to push through  
Carried away by a moonlight shadow  
Carried away by a moonlight shadow  
Far away on the other side.

Unfortunately, this topic has nothing to do with City Hunter.

# Chapter IV

## Exercise 00: Z

	Exercise 00
	Only the best know how to display Z
	Turn-in directory: <i>ex00/</i>
	Files to turn in: <b>z</b>
	Allowed functions: None

- Create a file called **z** that returns "Z", followed by a new line, whenever the **cat** command is used on it.

```
?>cat z
Z
?>
```



Google is your friend.

# Chapter V

## Exercise 01: testShell00

	Exercise 01
	What are attributes anyway ?
	Turn-in directory: <i>ex01/</i>
	Files to turn in: <b>testShell00.tar</b>
	Allowed functions: None

- Create a file called **testShell00** in your submission directory.
- Figure out a way to make the output look like this (except for the "total 1" line):

```
%> ls -l
total 1
-r--r-xr-x 1 XX XX 40 Jun 1 23:42 testShell00
%>
```

- Once you've achieved the previous steps, execute the following command to create the file to be submitted:

```
%> tar -cf testShell00.tar testShell00
```



- Don't worry about what appears instead of "XX".
- A year will be accepted instead of the time in the file's timestamp.



Did you check with your right-side neighbor?

# Chapter VI

## Exercise 02: Oh yeah, mooore...

	Exercise 02
	Oh yeah, mooore...
	Turn-in directory: <i>exo2/</i>
	Files to turn in: <i>exo2.tar</i>
	Allowed functions: None

- Create the following files and directories. Adjust their properties so that when you run the `ls -l` command in your directory, the output looks like this:

```
%> ls -l
total XX
drwx--xr-x 2 XX XX XX Jun 1 20:47 test0
-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1
dr-x---r-- 2 XX XX XX Jun 1 22:45 test2
-r-----r-- 2 XX XX 1 Jun 1 23:44 test3
-rw-r----x 1 XX XX 2 Jun 1 23:43 test4
-r-----r-- 2 XX XX 1 Jun 1 23:44 test5
1rwxrwxrwx 1 XX XX 5 Jun 1 22:20 test6 -> test0
%>
```

- Once you've completed this, run the following command to create the file to be submitted:

```
%> tar -cf exo2.tar *
```



- Don't worry about what appears instead of "XX".
- A year will be accepted instead of the time in the file's timestamp.



Don't hesitate to randomly ask someone in your cluster for help!

# Chapter VII

## Exercise 03: SSH me!

	Exercise 03
	SSH Key
	Turn-in directory: <i>ex03/</i>
	Files to turn in: <b>id_rsa_pub</b>
	Allowed functions: None

- Create your own SSH key. Once done:
  - Add your public key to your repository in a file named **id\_rsa\_pub**
  - Update your SSH key on the intranet. This will allow you to push the repository to our git server.



- The file name was not chosen randomly.
- Make sure you understand the difference between the public key and the private key.



Did you check with your left-side neighbor?

# Chapter VIII

## Exercise 04: midLS

	Exercise 04
	midLS
Turn-in directory: <i>ex04/</i>	
Files to turn in: <b>midLS</b>	
Allowed functions: None	

- In a file named midLS, write the command that lists all files and directories in your current directory (excluding hidden files or any file starting with a dot, including double dots).
- The output should be sorted by modification date, with entries separated by a comma and a space.
- Directory names should end with a slash (/).



Do only what is asked, nothing more!



- RTFM!
- Git push regularly.

# Chapter IX

## Exercise 05: GiT commit

	Exercise 05
	GiT commit?
	Turn-in directory: <i>ex05/</i>
	Files to turn in: <i>git_commit.sh</i>
	Allowed functions: None

- Create a shell script that displays the ids of the last 5 commits in your git repository.

```
%> bash git_commit.sh | cat -e
baa23b54f0adb7bf42623d6d0a6ed4587e11412a$
2f52d74b1387fa80ea844969e8dc5483b531ac1$
905f53d98656771334f53f59bb984fc29774701f$
5ddc8474f4f15b3fc72d08fc333e19c3a27078$
e94d0b448c03ec633f16d84d63beaef9ae7e7be8$
%>
```

- Your script will be tested in our own environment.



- RTFM!
- The first retry delay is short, trigger an intermediate evaluation to track your progress!

## Milestone Achieved, Keep Going!

You've completed the mandatory exercises for this project. Now, you have a choice:

- Continue with the **optional exercises** to explore more.
- Move on to your **next project**.

Both paths will introduce you to useful concepts. Consider the following before making your decision:

- Your first exam, as well as the end-of-week rush, will focus on C programming. It might therefore be useful to gain experience in this field beforehand. (You'll learn more about the rush soon).
- Your performance in this Piscine is evaluated on multiple factors:
  - Project completion is one aspect.
  - Overall progress through the full list of Piscine projects is another. Choose wisely to maximize your results.
- You can retry the same project in a few days or weeks until the end of the Piscine.
- Staying in sync with your peers promotes better collaboration.

# Chapter X

## Exercise 06: gitignore

	Exercise 06
	GiT
Turn-in directory:	<i>ex06/</i>
Files to turn in:	<code>git_ignore.sh</code>
Allowed functions:	None

- Write a short shell script that lists all the existing files ignored by your Git repository.

Example output:

```
%> bash git_ignore.sh | cat -e
.DS_Store$  
mywork.c~$  
%>
```

- Your script will be tested in our own environment.



- RTFM!
- Get inspired by others, but don't let them do your work!

# Chapter XI

## Exercise 07: diff

	Exercise 07
	Turn-in directory: <i>ex07/</i>
	Files to turn in: <b>b</b>
	Allowed functions: None

- Create a file named **b**, so that:

```
%>cat -e a
STARWARS$  
Episode IV, A NEW HOPE It is a period of civil war.$  
$  
Rebel spaceships, striking from a hidden base, have won their first victory against the evil  
Galactic Empire.$  
During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon,  
the DEATH STAR,$  
an armored space station with enough power to destroy an entire planet.$  
$  
Pursued by the Empire's sinister agents, Princess Leia races home aboard her starship, custodian  
of the stolen plans that can save her people and restore freedom to the galaxy...$  
$
```

```
%>diff a b > sw.diff
```



- `man patch`
- Don't blindly trust any source, always test, verify, and validate your results yourself!

# Chapter XII

## Exercise 08: clean

	Exercise 08
	Turn-in directory: <i>ex08/</i>
	Files to turn in: <b>clean</b>
	Allowed functions: None

- In a file called **clean** write a single command that:
  - Searches for all files in the current directory and its subdirectories that end with ~ (tilde) or, start and end with # (hash).
  - Displays the found files and deletes them.
- Only one command is allowed, no ';' or '&&' or other chaining tricks.



- **man find**
- **Collaboration is key to success!**

# Chapter XIII

## Exercise 09: Illusions, not tricks, Michael...

	Exercise 09
	Illusions, not tricks, Michael...
	Turn-in directory: <i>ex09/</i>
	Files to turn in: <b>ft_magic</b>
	Allowed functions: None

- Create a **magic file** named **ft\_magic** that is properly formatted to enable the **file** command to detect files of type **42 file**, defined as those containing the string "42" at the 42nd byte.



- **man file**
- Failure is part of your learning journey

# Chapter XIV

## Submission and peer-evaluation

Submit your assignment to your **Git** repository as usual. Only the work inside your repository will be evaluated during the defense. Make sure to double-check the filenames to ensure they are correct.



You must submit only the files explicitly required by the project instructions.