### VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY FACULTY OF COMPUTER SCIENCE AND ENGINEERING



## Course name h

# Report type h

# Report title h

Advisor: Advisor h



# Contents

1	Normal section	4
2	Better tables	4
3	Better enumerator	4
4	Codeblocks	5



#### 1 Normal section

This is how you normally work with LATEX, but you can also split a project into smaller files for easier management. To import other files, you can use \input{} or \include{}. There differences can be found at https://tex.stackexchange.com/a/250, but in short

\include{filename} = \clearpage \input{filename} \clearpage

#### 2 Better tables

The recommended way is by using the booktabs package and drop all vertical rules. Tabularx is simply tabular but with X environment, meaning that it will try to use all of \linewidth.

	OOP	FP
Pros		
Cons		

More information can be found at https://latex-tutorial.com/tables-in-latex/.

#### 3 Better enumerator

Normal enumerator gets the job done, but what if you want custom numbering? This implementation allows custom labeling, either by pre-defined rules or in-place.

a.yeah First item

b.yeah Second item

custom Third item



#### 4 Codeblocks

There are several ways to embed code in a  $\LaTeX$  file. Here are inline code, embedded codeblock, and external import.

• External import

```
class iostream:
    def __lshift__(self, other):
        print(other, end='')
        return self

def __repr__(self):
        return ''

if __name__ == "__main__":
        cout = iostream()
        endl = '\n'
        cout << "Hello" << ", " << "World!" << endl</pre>
```

• With custom line range

```
if __name__ == "__main__":
    cout = iostream()
    endl = '\n'
    cout << "Hello" << ", " << "World!" << endl</pre>
```

• Embedded

```
class iostream:

def __lshift__(self, other):

print(other, end='')

return self

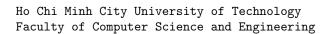
def __repr__(self):

return ''
```

• Inline

```
print('Hello, world!')
```

You can also define your custom inline as https://tex.stackexchange.com/a/148479. This is one way to input algorithms.





- 1 Initialize Q-table values (Q(s, a)) arbitrarily;
- 2 Initialize a state  $(s_t)$ ;
- ${f 3}$  Repeat Steps 4 to 6 until learning period ends;
- 4 Choose an action  $(a_t)$  for the current state  $(s_t)$  using an exploratory policy;
- **5** Take action  $(a_t)$  and observe the new state  $(s_t + 1)$  and reward  $(r_t + 1)$ ;
- ${f 6}$  Update Q-value;

Algorithm 1: QL algorithm



## References

- [1] Donald E. Knuth. Literate programming. The Computer Journal, 27(2):97–111, 1984.
- [2] Donald E. Knuth. The T<sub>E</sub>X Book. Addison-Wesley Professional, 1986.
- [3] Leslie Lamport. LaTeX: a Document Preparation System. Addison Wesley, Massachusetts, 2 edition, 1994.
- [4] Michael Lesk and Brian Kernighan. Computer typesetting of technical journals on UNIX. In *Proceedings of American Federation of Information Processing Societies: 1977 National Computer Conference*, pages 879–888, Dallas, Texas, 1977.
- [5] Frank Mittelbach, Michel Gossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LATEX Companion*. Addison-Wesley Professional, 2 edition, 2004.