Istanbul Technical University
Faculty of Computer and Informatics
Computer Engineering Department

BLG 100E
The Glorious LaTeX
Report

Group MadCoders
Besim Ongun Kanat - 150120047

December 2nd, 2016

Contents

1 Introduction

I made a LaTeXtemplate to help my friends on creating good looking reports.

2 General stuff

2.1 Text styles

You can make text **bold**, *italic*, underlined or in `typewriter fonts`. You can ***use*** `them` **`combined`**

You can make paragraphs centered

Or right aligned

The important paragraph: We can create titled paragraphs

2.2 Enumeration and lists

With the help of *enumitem* package we can create numbered lists as below:

1 Apples (We can use nested lists)

   A) Starking
   B) Golden

2 Kiwis

3 and of course Bananas!

We can also create unordered lists

- Ford Prefect
- Arthur Dent
- Zaphod Beeblebrox

2.3 Tables

Creating tables can become a bit annoying the [H] here ensures the table is displayed where it is defined.

Table 1: Table of great music

left column center aligned	center column right aligned	right column left aligned
I want to break free	We're the champions	Bohemian Rhapsody

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

For more info consult Wikibooks.
We can continue here but...

sometimes a clear page in our life is much better.

This page is left blank intentionally

3 Images and Figures

We can include images like:

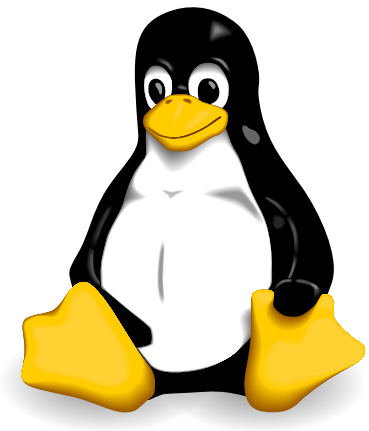
Figure 1: GTA

scale them relative to page width

Figure 2: GTA2

or even we can include PDFs (tip: Save SVG images as PDF) and they can scale (try to zoom in, it will not get pixelated)

Figure 3: Linux's mascot: Tux



4 Inserting Code Pieces

4.1 Pseudocode

Algorithm 1 The depth first search algorithm

```
Graph G
Node start
function DEPTH-FIRST-SEARCH(G, start)
    Tree T                                          ▷ The resulting search tree
    Stack S                                         ▷ An empty stack
    Set V                                           ▷ An empty set of visited nodes
    SET-ROOT(T,current)
    PUSH(S,start)
    while NOT-EMPTY(S) do
        current ← POP(S)
        if not CONTAINS(V, current) then
            INSERT(V, current)
            for all n : NEIGHBORS(current) do
                PUSH(S, n)
                INSERT-SUB-NODE(T, current, n)      ▷ Insert node to subtree of current
            end for
        end if
    end while
    return T
end function
```

4.2 Real Code

Code 1: Depth first search

```
class Graph
{
    set<int> nodes;
    vector< vector<int> > edge_list;
public:
    void dfs();
}
```