Chapter 1

**INTRODUCTION**

Currency exchange rates represent the value of one country's money in terms of another's. The value of different currencies is affected by the strength of the countries' economies as measured by various factors, including inflation, confidence in the government and interest rates in the country. For example, if one country is suffering from high inflation or has an unstable government, the value of its currency is likely to decrease. Understanding these factors can help you have a better idea of how exchanging your money works. Currency converters aim to maintain real-time information on current market or bank exchange rates, so that the calculated result changes whenever the value of either of the component currencies does.

STEP 1

Select the currency that you want to convert from and the currency that you want to convert to.

STEP 2

Choose Convert

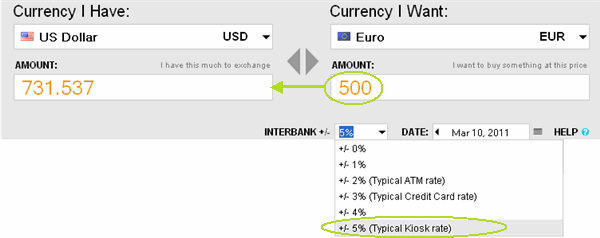


Fig 1.1 Converting Currencies

Chapter-2

**MATERIALS AND METHODS USED**

**2.1 Application Objective**

This currency converter converts 16 major currencies into one another.

**2.2 Anaconda**



Fig 2.2.1 Anaconda Symbol

Anaconda is a [freemium](https://en.wikipedia.org/wiki/Freemium) [open source](https://en.wikipedia.org/wiki/Open_source)distribution of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)) programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment. Package versions are managed by the [package management system](https://en.wikipedia.org/wiki/Package_manager) [*conda*](https://en.wikipedia.org/wiki/Conda_(package_manager)).

**2.3 Spyder**

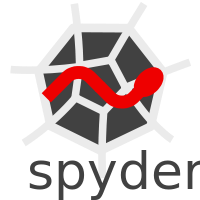


Fig2.3.1 Spyder Symbol

Spyder is a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features. Additionally, Spyder is a numerical computing environment thanks to the support of IPython and popular Python libraries such as NumPy, SciPy, or matplotlib. Spyder formerly known Pydee is an [open source](https://en.wikipedia.org/wiki/Open-source_software) cross-platform [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for scientific programming in the [Python language](https://en.wikipedia.org/wiki/Python_(programming_language)). Spyder integrates [NumPy](https://en.wikipedia.org/wiki/NumPy), [SciPy](https://en.wikipedia.org/wiki/SciPy), [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) and [IPython](https://en.wikipedia.org/wiki/IPython), as well as other open source software.[[4]](https://en.wikipedia.org/wiki/Spyder_(software)#cite_note-byu-4)[[5]](https://en.wikipedia.org/wiki/Spyder_(software)#cite_note-techworld-5) It is released under the [MIT license](https://en.wikipedia.org/wiki/MIT_license). Spyder is extensible with plug-in, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as [Pyflakes](https://en.wikipedia.org/w/index.php?title=Pyflakes&action=edit&redlink=1), [Pylint](https://en.wikipedia.org/wiki/Pylint) and Rope. It is available cross-platform through [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)), on Windows with WinPython and Python (x,y), on macOS through MacPorts, and on major Linux distributions such as [Arch Linux](https://en.wikipedia.org/wiki/Arch_Linux), [Debian](https://en.wikipedia.org/wiki/Debian), [Fedora](https://en.wikipedia.org/wiki/Fedora_(operating_system)), [Gentoo Linux](https://en.wikipedia.org/wiki/Gentoo_Linux), [openSUSE](https://en.wikipedia.org/wiki/OpenSUSE) and [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)). Spyder makes use of [Qt](https://en.wikipedia.org/wiki/Qt_(software)) either through the binding [PyQt](https://en.wikipedia.org/wiki/PyQt) or [PySide](https://en.wikipedia.org/wiki/PySide). This flexibility is reached through a small abstraction layer called QtPy.

**2.4 Tkinter**



Fig 2.4.1 Tkinter Symbol

Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of [Tk](http://www.tcl.tk/)inter. Tkinter is not the only [GuiProgramming](https://wiki.python.org/moin/GuiProgramming) toolkit for Python. It is however the most commonly used one. [CameronLaird](https://wiki.python.org/moin/CameronLaird) calls the yearly decision to keep Tkinter "one of the minor traditions of the Python world.

**2.5 Pandas**



**Fig 2.5.1 Pandas logo**

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the [Python](https://www.python.org/) programming language.

Pandas is a [NUMFocus](https://www.numfocus.org/open-source-projects.html) sponsored project. This will help ensure the success of development of pandas as a world-class open-source project, and makes it possible to [donate](https://pandas.pydata.org/donate.html) to the project.

pandas is a [Python](http://www.python.org/) package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

* Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
* Ordered and unordered (not necessarily fixed-frequency) time series data.
* Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
* Any other form of observational / statistical data sets. The data actually need not be labelled at all to be placed into a pandas data structure

What pandas does well:

* Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
* Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
* Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let *Series*, *DataFrame*, etc. automatically align the data for you in computations
* Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
* Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects
* Intelligent label-based slicing, fancy indexing, and sub setting of large data sets
* Intuitive merging and joining data sets
* Flexible reshaping and pivoting of data sets
* Hierarchical labelling of axes (possible to have multiple labels per tick)
* Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format
* Time series-specific functionality: date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging, etc.

## What problem does pandas solve?

Python has long been great for data mugging and preparation, but less so for data analysis and modelling. pandas helps fill this gap, enabling you to carry out your entire data analysis workflow in Python without having to switch to a more domain specific language like R.

Combined with the excellent [IPython](https://ipython.org/) toolkit and other libraries, the environment for doing data analysis in Python excels in performance, productivity, and the ability to collaborate.

pandas does not implement significant modelling functionality outside of linear and panel regression; for this, look to [stats models](http://statsmodels.sf.net/) and [scikit-learn](http://scikit-learn.org/). More work is still needed to make Python a first class statistical modelling environment, but we are well on our way toward that goal.

**2.6 Implementation**

currency\_converter (package imported) – code maker

Tkinter (standard python interface) - create GUI

Pandas (Libraries) - creates data frame

Packages imported:

import time

From currency\_converter import CurrencyConverter

import pandas as pd

import csv

from tables import createStandardTable as cst From the library currency\_converter the package CurrencyConverter is imported.

Another important feature:

Currency values change with changing Forex!

**2.7 FEATURES OF CURRENCY CONVERTER**

Excel Sheet is created and destroyed with various inputs given by the user & is displayed on the output.

Dynamically changing TIME.

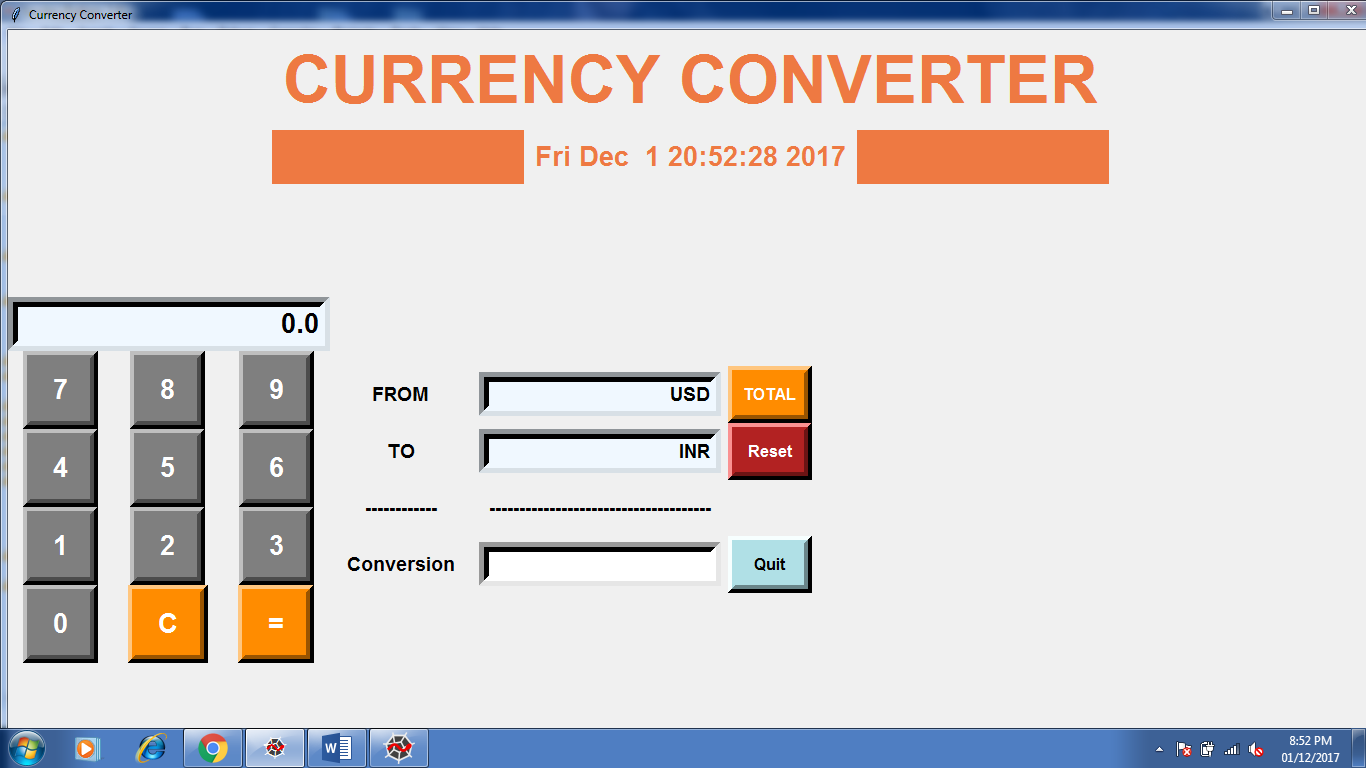
Creative Graphical User Interface.

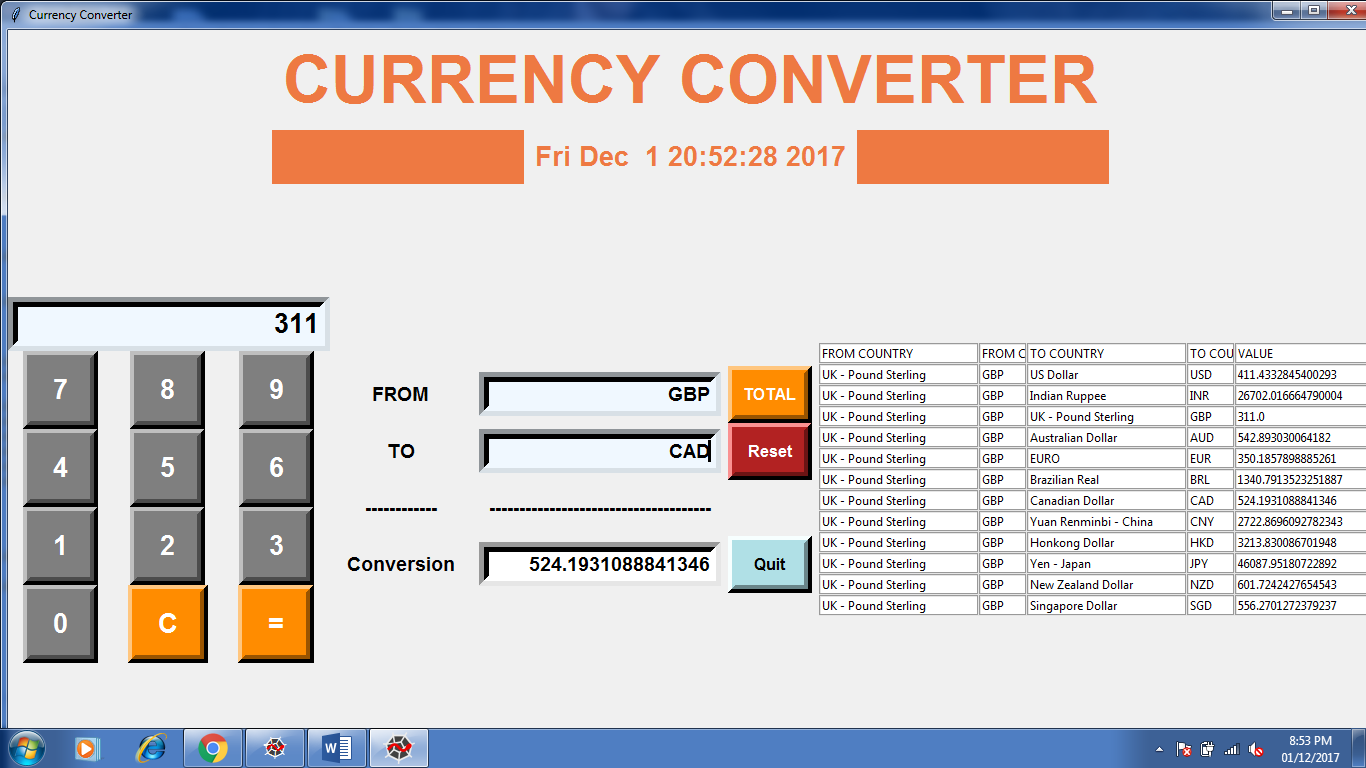
Number pad to take input values from the users.

Updated Value of Forex.

**Chapter-3**

**RESULT**



****

**Chapter-4**

**DISCUSSION**

A currency converter is software code that is designed to convert one [currency](https://en.wikipedia.org/wiki/Currency) into another in order to check its corresponding value. In order to convert one currency into another, a user enters an amount of money and chooses the currency she wishes to check the monetary value of. After that, the user selects one, or sometimes several other currencies, she would like to see the result in. The [application software](https://en.wikipedia.org/wiki/Application_software) then calculates and displays the corresponding amount of [money](https://en.wikipedia.org/wiki/Money). Currency converters aim to maintain real-time information on current market or bank exchange rates, so that the calculated result changes whenever the value of either of the component currencies does.

**REFERENCES**

[1] John V Guttag, Introduction to Computation and Programming using Python, Prentice Hall of India, 2013.

[2] Swaroop C H. , A Byte of Python

[3] Tkinter GUI Application Development Hotshot Book by Bhaskar Chaudhary

[4] Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython Book by Wes McKinney