KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Course Activity Report on

UNIVERSAL UNIT CONVERTER

Submitted in the partial fulfillment for the academic requirement of

VI Semester B.E.

in

PYTHON PROGRAMMING

Submitted by

NAME	USN
Anish Mayanache	2GI20EC021
Rohan Mangalore	2GI20EC101
Shraddha Karekar	2GI20EC126
Warren Viegas	2GI20ME087

GUIDE

Prof. Prasad Pujar

2022 – 2023

KARNATAK LAW SOCIETY'S GOGTE
INSTITUTE OF TECHNOLOGY
UDYAMBAG, BELAGAVI-590008
(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)
(APPROVED BY AICTE, NEW DELHI)

Department of Computer Science Engineering

CERTIFICATE

This is to certify that Master Anish Mayanache, Master Rohan Mangalore, Miss Shraddha Karekar and Master Warren Viegas of VI semester and bearing USN 2GI20EC021, 2GI20EC101,2GI20EC126 and 2GI20ME087 respectively, have satisfactorily completed the course activity (Seminar/Project) in Python Programming (18CS661). It can be considered as a bonafide work carried out in partial fulfillment for the academic requirement of VI Semester B.E. prescribed by KLS Gogte Institute of Technology, Belagavi during the academic year 2022- 2023.

The report has been approved as it satisfies the academic requirements in respect of Assignment (Course activity) prescribed for the said Degree.

Signature of the Faculty Member

Signature of the HOD

Date:

Marks allocation: (Page 2)

	Batch No. :					
1.	Seminar/Project Title:	Marks Range	USN/Roll No			
2.	Abstract (PO2)	0-2				
3.	Application of the topic to the course (PO2)	0-3				
4.	Literature survey and its findings (PO2)	0-4				
5.	Methodology, Results and Conclusion (PO1,PO3,PO4)	0-6				
6.	Report and Oral presentation skill (PO9,PO10)	0-5				
	Total	20				

*** 20 marks is converted to 10 marks for CGPA calculation**

CONTENTS:

PROBLEM STATEMENT:

Write a program to design a Universal Unit Converter which performs the below given conversions using user defined functions.

1. Length conversion (mm, cm, m, km, in, ft, yd, mi)
2. Time converter (s, min, h, day, week, month, year)
3. Currency converter (USD, EUR, GBP, JPY, INR)
4. Mass converter (mg, g, kg, lb, ton, oz)

OBJECTIVES OF THE GIVEN PROBLEM STATEMENT:

Learning the given program for implementing a universal unit converter in Python can help you achieve the following objectives:

1. Understanding Dictionary Data Structure: The program utilizes dictionaries to store conversion factors and rates for different units. By studying the program, you can gain knowledge of working with dictionaries, accessing values, and performing lookups.

2. Applying Mathematical Operations: The program involves performing mathematical operations such as multiplication and division to convert between units. It provides an opportunity to practice basic arithmetic operations in Python.

3. Handling User Input: The program prompts the user for input to specify the conversion type, values to convert, and units. It demonstrates techniques for accepting and processing user input, including handling different data types and validating user responses.

4. Implementing Modular Functions: The program divides the conversion logic into separate functions for each conversion type (length, time, currency, and mass). This approach promotes code modularity, reusability, and maintainability.

5. Building a Menu-Driven Program: The program employs a menu-driven approach to interact with the user. It allows the user to select the desired conversion type and provides appropriate prompts for input. Learning this approach helps in developing user-friendly programs with interactive menus.

6. Exploring Real-World Applications: The program models a universal unit converter, which is a practical application with various real-world use cases. Understanding the implementation details of such programs can be helpful in designing and developing similar applications.

7. Enhancing Problem-Solving Skills: The program addresses the problem of unit conversion, which requires analytical thinking, problem-solving, and algorithmic understanding. By studying the program, you can enhance your problem-solving skills and learn how to break down complex problems into smaller, manageable tasks.

8. Promoting Code Readability: The program follows good coding practices such as using meaningful variable names, proper indentation, and clear comments. Studying well-structured code enhances your understanding of writing readable and maintainable code.

PROGRAM CODE:

```
# Length conversion factors
length_factors = {
  "mm": 0.001,
  "cm": 0.01,
  "m": 1,
  "km": 1000,
  "in": 0.0254,
  "ft": 0.3048,
  "yd": 0.9144,
  "mi": 1609.34
}

# Time conversion factors
time_factors = {
  "sec": 1,
  "min": 60,
  "hr": 3600,
  "day": 86400,
  "week": 604800,
  "month": 2629800,
  "year": 31557600
}

# Currency conversion rates
currency_rates = {
  "USD": 1,
  "EUR": 0.92,
  "GBP": 0.79,
  "JPY": 144.31,
  "INR": 82.10
}

# Mass conversion factors
mass_factors = {
  "mg": 0.001,
  "g": 1,
  "kg": 1000,
  "oz": 28.3495,
  "lb": 453.592,
  "ton": 1000000
}
```

```
def convert_length(value, from_unit, to_unit):
  return value * length_factors[from_unit] / length_factors[to_unit]


def convert_time(value, from_unit, to_unit):
  return value * time_factors[from_unit] / time_factors[to_unit]


def convert_currency(value, from_currency, to_currency):
  return value * currency_rates[to_currency] / currency_rates[from_currency]


def convert_mass(value, from_unit, to_unit):
  return value * mass_factors[from_unit] / mass_factors[to_unit]


def main():
  while True:
    print("\n\nUniversal Unit Converter")
    print("==========================")
    print("1. Length Conversion")
    print("2. Time Conversion")
    print("3. Currency Conversion")
    print("4. Mass Conversion")
    choice = int(input("Enter your choice (1/2/3/4): "))

    if choice == 1:
      value = float(input("Enter the value to convert: "))
      from_unit = input(
        "Enter the source unit of length(mm,cm,km,m,in,ft,mi,yd): ")
      to_unit = input(
        "Enter the target unit of length(mm,cm,km,m,in,ft,mi,yd): ")
      result = convert_length(value, from_unit, to_unit)
      print("Converted value:", result)

    elif choice == 2:
      value = float(input("Enter the value to convert: "))
      from_unit = input(
        "Enter the source unit of time(sec, min, hr, day, week, month, year): "
      )
      to_unit = input(
        "Enter the target unit of time(sec, min, hr, day, week, month, year): "
      )
      result = convert_time(value, from_unit, to_unit)
      print("Converted value:", result)
```

```
 elif choice == 3:
      value = float(input("Enter the value to convert: "))
      from_currency = input("Enter the source currency(USD,EUR,GBP,JPY,INR): ")
      to_currency = input("Enter the target currency(USD,EUR,GBP,JPY,INR): ")
      result = convert_currency(value, from_currency, to_currency)
      print("Converted value:", result)

    elif choice == 4:
      value = float(input("Enter the value to convert: "))
      from_unit = input("Enter the source unit of mass(mg,g,kg,oz,lb,ton): ")
      to_unit = input("Enter the target unit of mass(mg,g,kg,oz,lb,ton): ")
      result = convert_mass(value, from_unit, to_unit)
      print("Converted value:", result)

    else:
      print("\n\n!!!!! INVALID CHOICE. TRY AGAIN !!!!!\n\n")


if __name__ == "__main__":
  main()
```

TEST CASES AND OUTPUTS:

```
Universal Unit Converter
========================
1. Length Conversion
2. Time Conversion
3. Currency Conversion
4. Mass Conversion
Enter your choice (1/2/3/4): 1
Enter the value to convert: 50
Enter the source unit of length(mm,cm,km,m,in,ft,mi,yd): mm
Enter the target unit of length(mm,cm,km,m,in,ft,mi,yd): cm
Converted value: 5.0
```

```
Universal Unit Converter
========================
1. Length Conversion
2. Time Conversion
3. Currency Conversion
4. Mass Conversion
Enter your choice (1/2/3/4): 2
Enter the value to convert: 2.5
Enter the source unit of time(sec, min, hr, day, week, month, year): hr
Enter the target unit of time(sec, min, hr, day, week, month, year): min
Converted value: 150.0
```

```
Universal Unit Converter
========================
1. Length Conversion
2. Time Conversion
3. Currency Conversion
4. Mass Conversion
Enter your choice (1/2/3/4): 3
Enter the value to convert: 40
Enter the source currency(USD,EUR,GBP,JPY,INR): USD
Enter the target currency(USD,EUR,GBP,JPY,INR): JPY
Converted value: 5772.4
```

```
Universal Unit Converter
========================
1. Length Conversion
2. Time Conversion
3. Currency Conversion
4. Mass Conversion
Enter your choice (1/2/3/4): 4
Enter the value to convert: 500
Enter the source unit of mass(mg,g,kg,oz,lb,ton): g
Enter the target unit of mass(mg,g,kg,oz,lb,ton): kg
Converted value: 0.5
```

```
Universal Unit Converter
========================
1. Length Conversion
2. Time Conversion
3. Currency Conversion
4. Mass Conversion
Enter your choice (1/2/3/4): 6


!!!!! INVALID CHOICE. TRY AGAIN !!!!!
```

CONCLUSION:

In conclusion, the Python program for implementing a universal unit converter provides a practical demonstration of various programming concepts and skills. By studying and understanding this program, you can achieve several objectives.

Furthermore, the program showcases effective techniques for handling user input, including validating and processing user responses. This knowledge is essential when developing interactive programs that require user interactions and inputs. The modular design of the program, with separate functions for each conversion type, promotes code modularity, reusability, and maintainability. This approach facilitates easier code management and enables you to extend the program by adding new conversion types in a structured manner.

Overall, the universal unit converter program serves as an excellent learning resource, enabling you to grasp and apply fundamental programming concepts, problem-solving techniques, user input handling, and code organization. The skills acquired through studying this program can be transferred to various programming scenarios and form a solid foundation for your programming journey.