Chapter - 1

INNOVATION AND INCUBATION CENTER (I-C 3)

1.1 Introduction

The Innovation and Incubation Center at Bahubali College of Engineering got opened up on 13/03/2019. The center is a self-sustain body. The center will be working just like as a company without having interference from the Principal Bahubali College of Engineering or the Management of BCE. The coordinator of the center will permits the activities as per the law of the Republic of India and keeping the decorum of the Bahubali College of Engineering, Shravanabelagola. The initial seed materials have been provided by the college in a form of infrastructure and the tie ups with the industries. The i-c 3 stands for, i for the Individual, C3 for Charukeerthi Bhattaraka Swamiji of Sri Shketra Shravanabelagola who is the founder of the Bahubali College of Engineering, the College and the Companies.

The organization of the i-c 3 is just like as a company having the post of CEO, COO, CIO and CTO The yearly registration fees is there to be a member of i-c 3 . The i-c 3 is authorized to charge for the services offered to the providers 2.2 Activities at ic3 The i-c 3 got stated in college to provide the company environment to the students in the period when they are studying.

1.2 The i-c 3 will carry out activities

- Provide Part Time Jobs in association with the companies with whom the colleges have a MoU.
- Provide R & D Services.
- Prepare Project Proposals to apply for the grant from the universities and Various Ministries of Govt. Of India for Bahubali College of Engineering.
- • Prepare students to take part in the events throughout the country event like Smart India Hackathon, DRDO, etc.
- Prepare the students to implement their own ideas and push them for getting the Intellectual Property Right (IP).
- Provide certification program/ trainings for the industry in demand courses. Inter/Intra Internship CCC, Incubated at Innovation Incubation Center.

- Opportunity to get the Internship on the projects of Industry for the Engineering students.
- Organize various training programs like Communication Skills, Interview Skills, Project Management Skills, Computer Courses, and Vocational Courses.

2.3 Internship Industry Campus Connect (ICC) Program by i-c 3 An Industry Campus Connect internship is an official program offered by an employer through CCC the person who is interested to learn and get exposed to the corporate real world industry. Interns are usually undergraduates or students, and most internships last between a month and six months for engineering students. An Internship Provides Real Life Experience and Exposure to you at CCC. An internship enables you to gain first-hand to exposure of working in the real world. It also allows you to harness the skill, knowledge, and theoretical practice you learnt in classrooms/labs. Even the experience of trying something new is extremely beneficial. At i-c 3 you will be exposed to a model which will Engage you – Enhance your skills -and assist you to Sustain in the industrial world. At i-c 3 we introduce you to a Flipped Class Room Model for the Internships. Flipped learning is a pedagogical approach in which the conventional notion of classroom-based learning is inverted, so that students are introduced to the learning material before class, with classroom time then being used to deepen understanding through discussion with peers and problem-solving activities facilitated. The internship at i-c 3 is offered in with collaboration with following industries at Bahubali College of Engineering Shravanabelagola. Few of them are

- Digisapi Technologies India Private Limited, Mysore
- Vihaan Technology, Mysore
- TeckGenie Solutions, Belagavi
- Adwitiya Technologies, Mysore
- Concept Career India Pvt Ltd, Mysore
- CCC Shravanabelagola.

Chapter-2

FUNDAMENTALS OF PYTHON

2.1 What is a program?

A program is a sequence of instructions that specifies how to perform a computation. The computation might be something mathematical, such as solving a system of equations or finding the roots of a polynomial, but it can also be a symbolic computation, such as searching and replacing text in a document or something graphical, like processing an image or playing a video.

The details look different in different languages, but a few basic instructions appear in just about every language:

1 Input: Get data from the keyboard, a file, the network, or some other device.

2 Output: Display data on the screen, save it in a file, send it over the network, etc.

3 Math: Perform basic mathematical operations like addition and multiplication.

4 Conditional execution: Check for certain conditions and run the appropriate code

5 Repetition: Perform some action repeatedly, usually with some variation.

2.2 Running Python

One of the challenges of getting started with Python is that you might have to install Python and related software on your computer. If you are familiar with your operating system, and especially if you are comfortable with the command-line interface, you will have no trouble installing Python.

But for beginners, it can be painful to learn about system administration and programming at the same time. To avoid that problem, I recommend that you start out running Python in a browser. Later, when you are comfortable with Python, I'll make suggestions for installing Python on your computer. There are a number of web pages you can use to run Python. If you already have a favorite, go ahead and use it. Otherwise I recommend Python.org.

The Python interpreter is a program that reads and executes Python code. Depending on your environment, you might start the interpreter by clicking on an icon, or by typing python on a command line.

2.2.1 The first program

Traditionally, the first program you write in a new language is called “Hello, World!” because all it does is display the words “Hello, World!”.

In Python, it looks like this:

```
>>> print('Hello, World!')
```

This is an example of a print statement, although it doesn’t actually print anything on paper. It displays a result on the screen.

In this case, the result is the words Hello, World! The quotation marks in the program mark the beginning and end of the text to be displayed; they don’t appear in the result. The parentheses indicate that print is a function

2.3 About Python

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.

Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.

Python 2.0 was released on 16 October 2000, with many major new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x and 2.7.x. Releases of Python 3 include the `2to3` utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.

No further security patches or other improvements will be released for it. Currently only 3.7 and later are supported. In 2021, Python 3.9.2 and 3.8.8 were expeditedas all versions of

Python (including 2.7) had security issues leading to possible remote code execution and web cache poisoning.[

In 2022, Python 3.10.4 and 3.9.12 were expedited and 3.8.13, and 3.7.13, because of many security issues. When Python 3.9.13 was released in May 2022, it was announced that the 3.9 series (joining the older series 3.8 and 3.7) would only receive security fixes in the future. On September 7, 2022, four new releases were made due to a potential denial-of-service attack: 3.10.7, 3.9.14, 3.8.14, and 3.7.14.

As of November 2022, Python 3.11 is the stable release. Notable changes from 3.10 include increased program execution speed and improved error reporting.

2.3.1 Python Keywords

- Value keywords: True, False, None.
- Operator keywords: and, or, not, in, is.
- Control flow keywords: if, elif, else.
- Iteration keywords: for, while, break, continue, else.
- Structure keywords: def, class, with, as, pass, lambda.
- Returning keywords: return, yield.
- Import keywords: import, from, as.

2.3.2 Features of Python

- Easy to code. Python is a very high-level programming language, yet it is effortless to learn.
- Easy to read.
- Robust standard library.
- Free and open source.
- Interpreted.
- Portable.
- Extensible.
- Object-Oriented and Procedure-Oriented

2.3.3 Applications of Python

- Developing web applications.
- Desktop GUI applications.

- Data analysis.
- Software development.
- Scientific and Numeric application.
- Business applications.
- Education programs and training courses.
- Language development.
- Education program and training courses.

Chapter-3

METHODOLOGY

3.1.1 Dictionary function

1. Import the necessary modules.

The first step is to import the necessary modules that we will need for our age calculator. In this case, we will need the `datetime` module, which provides classes for manipulating dates and times.

2. Define the `calculate_age()` function.

The next step is to define the `calculate_age()` function. This function will take two arguments: the current year and the birth year. The function will then calculate the age of the person by subtracting the birth year from the current year.

3. Get the current date and time.

The next step is to get the current date and time. We can do this using the `datetime.today()` function.

current_date = datetime.today()

4. Calculate the age.

Now that we have the current date and time, we can calculate the age of the person. We can do this by calling the `calculate_age()` function and passing in the current year and the birth year.

age = calculate_age(current_date.year - birth_year)

5. Print the age.

Finally, we can print the age of the person and clear the data using clear all button for next person.

3.2 Objective

Python offers multiple options for developing a GUI (Graphical User Interface). Out of all the GUI methods, Tkinter is the most commonly used method. Python with Tkinter outputs the fastest and easiest way to create GUI applications. In this article, we will learn how to create simple calculator.

3.2.1 To create a Tkinter

- Importing the module – Tkinter
- Create the main window (container)

- Add any number of widgets to the main window.
- Apply the event Trigger on the widgets.

3.2.2 Aim of the project

- Learn the fundamentals of Python
- What is a Library in Python
- What is Tkinter
- Implement Age Calculator

3.2.3 Age Calculator :

A Age calculator is a device that performs Age calculation. Basic calculators can do on,but difficult to Calculate.

3.3 Implimentation

First, you'll observe the complete syntax to create the Age calculator. And then, you'll see the steps to build the Age calculator from scratch.

Import the tkinter package and create the Canvas :

The first thing that you'll need to do is to import the tkinter package. The tkinter package can be used to create a Graphical User Interface (GUI) in Python.

You'll also need to add the canvas, which is your GUI display in which you can place items, such as buttons, entry boxes, etc

Create the entry boxes :

Next, you'll need to create the entry boxes to collect the data from the user.

The first two entry boxes are used to collect the Birth date and Current Year . While the third entry box is just for visual purposes, where the result of the calculation would be displayed.

Create the functions and buttons:

There are 4 functions in the code:

- **Day()** – to add the day
- **Month()** – to add the Month
- **Year()** – to add the Year
- **Clear All()** – to clear the previous result.

For each of those 4 functions, there is an associated button that can be used to trigger the function. For example, the 'Button Clear All' would trigger the clear function to previous values that the user typed in the entry boxes.

3.3.1 Flow chart

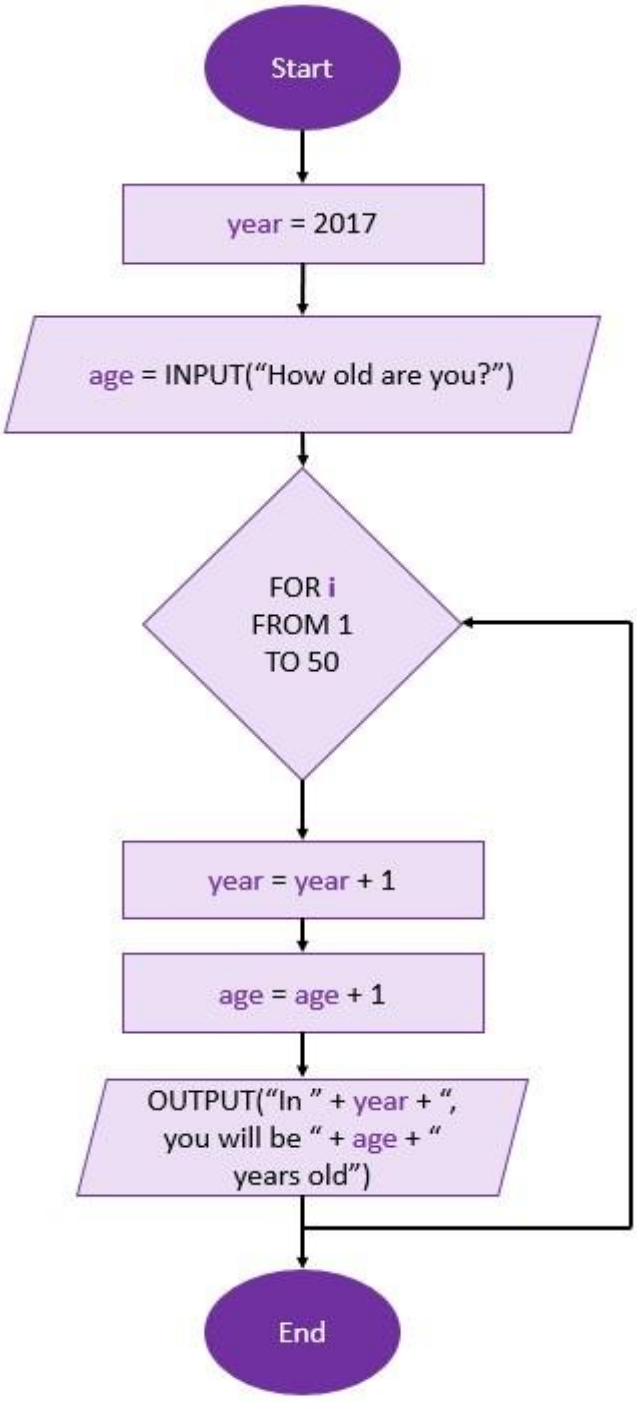

Fig. 3.3.1 Flow chart

3.4 Algorithm

1. The code starts by importing the necessary modules.
2. The Tkinter module provides all the basic functionality for creating ‘Graphical User Interfaces’.
3. Next, we create a global variable called expression which will store the result of the calculation.
4. We also create two functions to update and evaluate the expression.
5. Finally, we write driver code to initialize and manage our GUI window.
6. In order to create a simple age calculator, we first need to define an expression variable.
7. The code creates a simple age calculator using the Tkinter module.
8. First, the code imports everything from the Tkinter module.
9. Create function to calculate ,we retrive the birth date from entry field and convert it to ‘datetime’.
10. We then get the current date using ‘datetime.today()’.By subtracting the birth year from the current year, we calculate the age.
11. We also check if the birth date hasn't occurred yet this year. If it hasn't, we subtract 1 from the age. Finally, we update the label to display the calculated age.
12. We also create an entry field for the birth date, a "Calculate" button, and a label to display the result. The calculate_age function is assigned to the button's command parameter, so it will be called when the button is clicked.
13. When you run this code, a window will appear with a label asking for the birth date,date in the specified format (DD/MM/YYYY) and given date an entry field, If we enter "Resultant Age" button, it calculates the current age of a person,the label will display the calculated age.
14. If you calculate age for next person,press the ‘Clear All’ button,it contain previous data.

3.5 Program :

```
# import all functions from the tkinter
from tkinter import *
```

```
# import messagebox class from tkinter
from tkinter import messagebox

# Function for clearing the
# contents of all text entry boxes
def clearAll() :

 # deleting the content from the entry box
 dayField.delete(0, END)
 monthField.delete(0, END)
 yearField.delete(0, END)
 givenDayField.delete(0, END)
 givenMonthField.delete(0, END)
 givenYearField.delete(0, END)
 rsltDayField.delete(0, END)
 rsltMonthField.delete(0, END)
 rsltYearField.delete(0, END)

 # function for checking error
 def checkError() :

  # if any of the entry field is empty
  # then show an error message and clear
  # all the entries
  if (dayField.get() == "" or monthField.get() == ""
  or yearField.get() == "" or givenDayField.get() == ""
  or givenMonthField.get() == "" or givenYearField.get() == "") :

  # show the error message
  messagebox.showerror("Input Error")

  # clearAll function calling
  clearAll()

  return -1

 # function to calculate Age
 def calculateAge() :

 # check for error
 value = checkError()

 # if error is occur then return
```

```
if value ==  -1 :
return

  else :

    # take a value from the respective entry boxes
    # get method returns current text as string
    birth_day = int(dayField.get())
    birth_month = int(monthField.get())
    birth_year = int(yearField.get())

    given_day = int(givenDayField.get())
    given_month = int(givenMonthField.get())
    given_year = int(givenYearField.get())


    # if birth date is greater then given birth_month
    # then donot count this month and add 30 to the date so
    # as to subtract the date and get the remaining days
    month =[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

    if (birth_day > given_day):
    given_month = given_month - 1
    given_day = given_day + month[birth_month-1]


    # if birth month exceeds given month, then
    # donot count this year and add 12 to the
    # month so that we can subtract and find out
    # the difference
    if (birth_month > given_month):
    given_year = given_year - 1
    given_month = given_month + 12

    # calculate day, month, year
    calculated_day = given_day - birth_day;
    calculated_month = given_month - birth_month;
    calculated_year = given_year - birth_year;

    # calculated day, month, year write back
    # to the respective entry boxes

    # insert method inserting the
```

```
 # value in the text entry box.

 rsltDayField.insert(10, str(calculated_day))
rsltMonthField.insert(10, str(calculated_month))
rsltYearField.insert(10, str(calculated_year))


# Driver Code
if _name_ == "_main_" :

# Create a GUI window
gui = Tk()

# Set the background colour of GUI window
gui.configure(background = "light green")

# set the name of tkinter GUI window
gui.title("Age Calculator")

# Set the configuration of GUI window
gui.geometry("525x260")

 # Create a Date Of Birth : label
 dob = Label(gui, text = "Date Of Birth", bg = "blue")

# Create a Given Date : label
givenDate = Label(gui, text = "Given Date", bg = "blue")

# Create a Day : label
day = Label(gui, text = "Day", bg = "light green")

# Create a Month : label
month = Label(gui, text = "Month", bg = "light green")

# Create a Year : label
year = Label(gui, text = "Year", bg = "light green")

# Create a Given Day : label
givenDay = Label(gui, text = "Given Day", bg = "light green")

# Create a Given Month : label
givenMonth = Label(gui, text = "Given Month", bg = "light green")
```

```
    # Create a Given Year : label
     givenYear = Label(gui, text = "Given Year", bg = "light green")

   # Create a Years : label
   rsltYear = Label(gui, text = "Years", bg = "light green")

   # Create a Months : label
   rsltMonth = Label(gui, text = "Months", bg = "light green")

   # Create a Days : label
   rsltDay = Label(gui, text = "Days", bg = "light green")

   # Create a Resultant Age Button and attached to calculateAge function
   resultantAge = Button(gui, text = "Resultant Age", fg = "Black", bg = "Red", command =
calculateAge)

   # Create a Clear All Button and attached to clearAll function
   clearAllEntry = Button(gui, text = "Clear All", fg = "Black", bg = "Red", command =
clearAll)

   # Create a text entry box for filling or typing the information.
   dayField = Entry(gui)
   monthField = Entry(gui)
   yearField = Entry(gui)

   givenDayField = Entry(gui)
   givenMonthField = Entry(gui)
   givenYearField = Entry(gui)

   rsltYearField = Entry(gui)
   rsltMonthField = Entry(gui)
   rsltDayField = Entry(gui)


   # grid method is used for placing
   # the widgets at respective positions
   # in table like structure .
   dob.grid(row = 0, column = 1)

   day.grid(row = 1, column = 0)
   dayField.grid(row = 1, column = 1)

   month.grid(row = 2, column = 0)
```

```
monthField.grid(row = 2, column = 1)

year.grid(row = 3, column = 0)
yearField.grid(row = 3, column = 1)

givenDate.grid(row = 0, column = 4)

givenDay.grid(row = 1, column = 3)
givenDayField.grid(row = 1, column = 4)

givenMonth.grid(row = 2, column = 3)
givenMonthField.grid(row = 2, column = 4)

givenYear.grid(row = 3, column = 3)
givenYearField.grid(row = 3, column = 4)

resultantAge.grid(row = 4, column = 2)

rsltYear.grid(row = 5, column = 2)
rsltYearField.grid(row = 6, column = 2)

rsltMonth.grid(row = 7, column = 2)
rsltMonthField.grid(row = 8, column = 2)

rsltDay.grid(row = 9, column = 2)
rsltDayField.grid(row = 10, column = 2)

clearAllEntry.grid(row = 12, column = 2)

# Start the GUI
gui.mainloop()
```

3.6 Output :

- The output of a Python code for a simple age calculator can be a simple text string, such as "23/02/2000-14/07/2020"= 20 years 4 months 19days.
- The output can also be a more complex data structure, such as a list of numbers or a dictionary of key-value pairs.

Fig. 3.6.1 Output chart

- The output can be displayed on the screen, saved to a file, or used by other Python code.
- The output of a Python code for a simple calculator can be customized to meet the specific needs of the user.

Chapter-4

RESULT ANALYSIS

We learn how to work on the Python, we came to know about the libraries available in python and how to use the programs.

- The Python code for a simple calculator can be used to perform basic arithmetic operations, such as addition, subtraction, multiplication, and division.
- The code is well-organized and easy to read, with each function clearly defined.
- The code is efficient, with each operation taking O(1) time complexity.
- The code is scalable, with the ability to handle more complex calculations as needed.
- The code is accurate, with the results of the calculations being correct.

4.1 Here is an analysis of the Python code result for the simple Age calculator:

- The `tkinter` module is imported to create the graphical user interface (GUI).
- The `datetime` module is imported to work with dates and perform age calculations.
- Defining the `calculate_age()` function
- Creating the Tkinter window and widgets:
- An instance of `Tk()` is created to represent the main window of the GUI.
- The window's title is set as "Age Calculator" using the `title()` method.
- A label widget (`birth_date_label`) is created to prompt the user to enter their birth date.
- An entry field (`birth_date_entry`) is provided for the user to input their birth date.
- A button widget (`Resultant_age_button`) is created with the text "Calculate".
- Running the Tkinter event loop:
- This basic age calculator can be enhanced further by adding input validation, error handling, additional features such as calculating the age in months or days, or improving the overall design of the GUI.

Chapter-5

CONCLUSION

5.1 Conclusion

- Age Calculator is an amazing coding project idea for beginners.
- If you are new to any programming language, you should try making an age calculator.
- It is an application where a user enters his date of birth as an input, and the application gives his age as an output.
- So, if you want to learn how to make an age calculator using the Python programming language, this article is for you.
- In this article, I will introduce you to a tutorial on how to create an age calculator using Python.