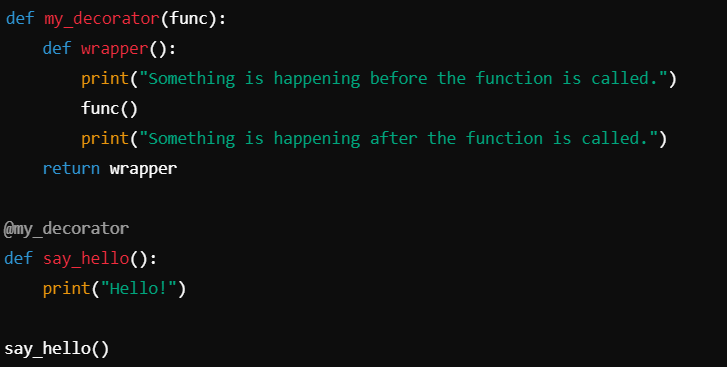
1. What are decorators in python? Provide a code of same

A decorator in Python is a function that takes another function as an argument and extends its behavior without explicitly modifying it. They are often used for logging, enforcing access control, instrumentation, and more.

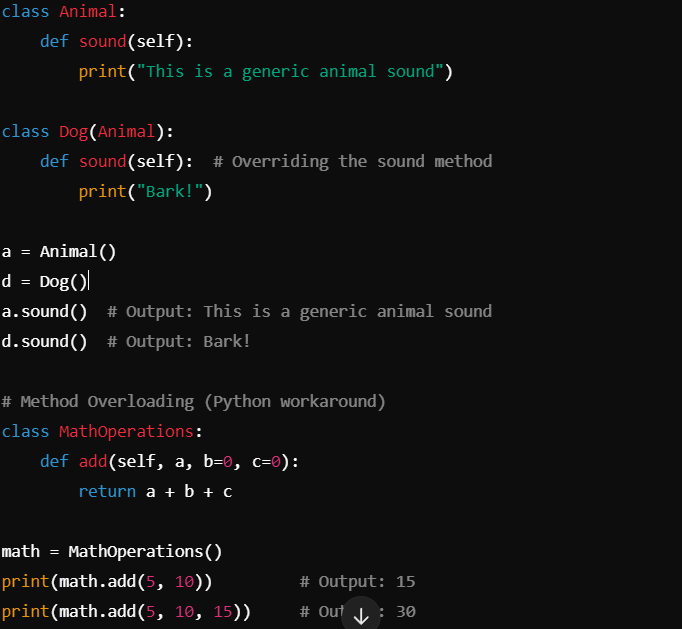
Example Code



2. What do you understand by method overriding and method overloading? Give the complete implementation

Method Overriding: This occurs when a subclass provides a specific implementation of a method that is already defined in its superclass. It is useful for runtime polymorphism.

Method Overloading: Python does not support traditional method overloading. However, we can achieve similar behavior by setting default values or using variable arguments.



3. Code for hybrid inheritance



4.Explain about the time complexity and space complexity of stacks, queues, Linekdin list

Stack:

Time Complexity:

Push, Pop, Peek: O(1)

Space Complexity:

O(n) for n elements.

Queue:

Time Complexity:

Enqueue, Dequeue: O(1)

Space Complexity:

O(n) for n elements.

Linked List:

Time Complexity:

Insert/Delete at head: O(1)

Insert/Delete at tail: O(n)

Search: O(n)

Space Complexity:

O(n) for n nodes.

5.Explain any 3 widgets in Streamlit

st.button(): Adds a button to the app, which the user can click to trigger an action.

st.slider(): Allows users to select a numeric value from a range.

st.selectbox(): Displays a dropdown menu from which the user can select an option.

6. Explain session in Streamlit

In Streamlit, session state allows you to store data across user interactions. This is useful for tracking user input or app behavior as users navigate. You can use st.session\_state to create and update variables that retain values as the user interacts with the app.

7. How will you run any file in Streamlit

To run a file in Streamlit, navigate to the directory containing your .py file and use the command:  
streamlit run your\_script.py

8. Plot graph such as bar chart, pie chart

Bar Chart:

import streamlit as st

import pandas as pd

data = pd.DataFrame({

'Category': ['A', 'B', 'C'],

'Values': [10, 20, 15]

})

st.bar\_chart(data.set\_index('Category'))

Pie Chart (using Plotly):

import streamlit as st

import plotly.express as px

labels = ['A', 'B', 'C']

values = [10, 20, 15]

fig = px.pie(values=values, names=labels)

st.plotly\_chart(fig)