Mod3

1.Turtle program to draw a shape (Parallelogram,square,hexagon)with fill color?

\*\* any shape can be asked

import turtle  
t=turtle.Turtle()  
t.width(5)  
t.begin\_fill()  
t.forward(150)  
t.left(45)  
t.forward(150)  
t.left(135)  
t.forward(150)  
t.left(45)  
t.forward(150)  
t.fillcolor("blue")  
t.end\_fill()

import turtle

t = turtle.Turtle()

t.width(5)

# Start filling the square

t.begin\_fill()

# Set the fill color

t.fillcolor("blue")

# Draw the square

for \_ in range(4):

t.forward(150)

t.left(90)

# End filling

t.end\_fill()

import turtle  
t = turtle.Turtle()  
t.width(5)  
# Start filling the hexagon  
t.begin\_fill()  
# Set the fill color  
t.fillcolor("blue")  
# Draw the hexagon  
for \_ in range(6):  
 t.forward(150)  
 t.left(60)  
# End filling  
t.end\_fill()

2.Explain the steps to create GUI using Tkinter

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

3.Difference between terminal based and GUI based programming ?

A screenshot of a computer program

AI-generated content may be incorrect.

Or

A black and white screen with white text

AI-generated content may be incorrect.

4.Image processing methods ?

A screenshot of a computer program

AI-generated content may be incorrect.

5. program to convert an color image to black &white

from images import Image  
  
  
def blackAndWhite(image):  
 *"""Converts the argument image to black and white."""* blackPixel = (0, 0, 0)  
 whitePixel = (255, 255, 255)  
  
 for y in range(image.getHeight()):  
 for x in range(image.getWidth()):  
 (r, g, b) = image.getPixel(x, y)  
 average = (r + g + b) // 3  
 if average < 128:  
 image.setPixel(x, y, blackPixel)  
 else:  
 image.setPixel(x, y, whitePixel)  
  
  
def main(filename="smokey.gif"):  
 image = Image(filename)  
 print("Close the image window to continue.")  
 image.draw()  
 blackAndWhite(image)  
 print("Close the image window to quit.")  
 image.draw()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

6. program to Resize an image ?

from PIL import Image

# Open the image

img = Image.open("input.jpg")

# Resize the image to 200x200

img = img.resize((200, 200))

# Save the resized image

img.save("resized.jpg")

print("Image resized successfully!")

7.design a gui for age calculation using python program

from breezypythongui import EasyFrame  
from datetime import date  
  
today = date.today()  
  
  
class Birthday(EasyFrame):  
  
 def \_\_init\_\_(self):  
 *"""Sets up the window and widgets."""* EasyFrame.\_\_init\_\_(self, title="Age Calculator", height=500, width=500, background="orange")  
 # Label and field for the input  
 self.addLabel(text="Enter birth year",  
 row=0, column=0)  
 self.inputField1 = self.addIntegerField(value=0,  
 row=0,  
 column=1,  
 width=10)  
 self.addLabel(text="Enter birth month",  
 row=1, column=0)  
 self.inputField2 = self.addIntegerField(value=0,  
 row=1,  
 column=1,  
 width=10)  
 self.addLabel(text="Enter birth date",  
 row=2, column=0)  
 self.inputField3 = self.addIntegerField(value=0,  
 row=2,  
 column=1,  
 width=10)  
 self.addButton(text="Calculate Age", row=4, column=0,  
 columnspan=2, command=self.birth\_calc)  
 self.addLabel(text="Your Age is ",  
 row=6, column=0)  
 self.outputField = self.addIntegerField(value=0,  
 row=6,  
 column=1,  
 width=10,  
 state="readonly")  
  
 # The event handling method for the button  
 def birth\_calc(self):  
 d = self.inputField3.getNumber()  
 m = self.inputField2.getNumber()  
 y = self.inputField1.getNumber()  
 age = today.year - y - ((today.month, today.day) < (m, d))  
 self.outputField.setNumber(age)  
  
  
# Instantiate and pop up the window."""  
if \_\_name\_\_ == "\_\_main\_\_":  
 Birthday().mainloop()

8.convert degree Celsius to Fahrenheit using GUI

from breezypythongui import EasyFrame

class TemperatureConverter(EasyFrame):

def \_\_init\_\_(self):

"""Sets up the window and widgets."""

EasyFrame.\_\_init\_\_(self, title="Celsius to Fahrenheit Converter", height=300, width=300, background="lightblue")

# Label and input field for Celsius

self.addLabel(text="Enter temperature in Celsius:", row=0, column=0)

self.inputField = self.addFloatField(value=0.0, row=0, column=1, width=10)

# Button to trigger conversion

self.addButton(text="Convert", row=1, column=0, columnspan=2, command=self.convert\_temp)

# Label and output field for Fahrenheit

self.addLabel(text="Temperature in Fahrenheit:", row=2, column=0)

self.outputField = self.addFloatField(value=0.0, row=2, column=1, width=10, state="readonly")

# The event handling method for the button

def convert\_temp(self):

celsius = self.inputField.getNumber()

fahrenheit = (celsius \* 9/5) + 32

self.outputField.setNumber(fahrenheit)

# Instantiate and pop up the window

if \_\_name\_\_ == "\_\_main\_\_":

TemperatureConverter().mainloop()

Mod4

1.explain Self argument in class

A close-up of a text

AI-generated content may be incorrect.

2.Explain \_\_init\_\_ in class.

A text on a white background

AI-generated content may be incorrect.

A close-up of a card

AI-generated content may be incorrect.

3.simple programs using class

a. area and perimeter of a rectangle using class

class Rectangle:

def \_\_init\_\_(self, length, width):

self.length = length

self.width = width

def area(self):

return self.length \* self.width

def perimeter(self):

return 2 \* (self.length + self.width)

# Creating an object of the class

rect = Rectangle(10, 5)

print("Area:", rect.area())

print("Perimeter:", rect.perimeter())

b.Bank account class and methods to calculate balance and display balance

class BankAccount:

def \_\_init\_\_(self, balance=0):

self.balance = balance

def deposit(self, amount):

self.balance += amount

print(f"Deposited: ${amount}")

def withdraw(self, amount):

if amount > self.balance:

print("Insufficient balance!")

else:

self.balance -= amount

print(f"Withdrawn: ${amount}")

def display\_balance(self):

print(f"Current Balance: ${self.balance}")

# Creating an object of the class

account = BankAccount(1000)

account.deposit(500)

account.withdraw(300)

account.display\_balance()

c. complex number arithmetic

class ComplexNumber:

def \_\_init\_\_(self, real, imag):

self.real = real

self.imag = imag

def add(self, other):

return ComplexNumber(self.real + other.real, self.imag + other.imag)

def subtract(self, other):

return ComplexNumber(self.real - other.real, self.imag - other.imag)

def display(self):

print(f"{self.real} + {self.imag}i")

# Creating objects

num1 = ComplexNumber(2, 3)

num2 = ComplexNumber(4, 5)

# Performing operations

result\_add = num1.add(num2)

result\_sub = num1.subtract(num2)

# Display results

print("Addition:", end=" ")

result\_add.display()

print("Subtraction:", end=" ")

result\_sub.display()

4.operator overloading programs

a. complex number addition using operator overloading

class ComplexNumber:

def \_\_init\_\_(self, real, imag):

self.real = real

self.imag = imag

# Overloading the '+' operator

def \_\_add\_\_(self, other):

return ComplexNumber(self.real + other.real, self.imag + other.imag)

# For displaying the result

def \_\_str\_\_(self):

return f"{self.real} + {self.imag}i"

# Creating objects

num1 = ComplexNumber(2, 3)

num2 = ComplexNumber(4, 5)

# Adding using overloaded '+' operator

result = num1 + num2

print("Addition of Complex Numbers:", result)

b.compare number using ‘==’ operator

class Number:

def \_\_init\_\_(self, value):

self.value = value

# Overloading the '==' operator

def \_\_eq\_\_(self, other):

return self.value == other.value

# Creating objects

num1 = Number(10)

num2 = Number(10)

num3 = Number(20)

# Comparing the numbers

print(num1 == num2) # True

print(num1 == num3) # False

5.explain abstract class and abstract methods with an example

A screenshot of a computer screen

AI-generated content may be incorrect.A screen shot of a program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

6.Exception handling in Python

A close-up of a document

AI-generated content may be incorrect.

A close-up of a text

AI-generated content may be incorrect.

A white paper with text on it

AI-generated content may be incorrect. A text on a white background

AI-generated content may be incorrect.