**Name: Bryan Bibb**

Instructions:

Answer the following questions in this document. Do NOT change the question order, numbering, or wording. Submit your completed Journal Entry Assignment via Blackboard. Your Journal Entries are part of the Quiz and Assignments grading category in CPT187 which counts as 15% of your final grade. Refer to the CPT187 Course Syllabus in Blackboard for more information about how your grade in this class is calculated.

IMPORTANT: Please provide code examples whenever possible in your responses to these questions.  
  
Please enter your responses in BLUE so your instructor can see them better. Thanks.

Define with examples the following terms, methods, components, and arguments as explained in Chapter 18:

1. askyesno()

method of the messagebox module which displays a message box asking for confirmation with yes and no buttons. Returns True if 'yes' and False if 'no'.

messagebox.askyesno("Are you sure?")

1. bind

permanently associates a variable with a StringVar object, used by the textvariable argument of the Entry() constructor. When the user enters text it gets saved in the object that was created, so you can use get() to retrieve the user text.

investmentText = tk.StringVar # creates the StringVar object

# binds the StringVar to the text entry field

investmentEntry = ttk.Entry(frame, width=25, textvariable=investmenText

1. button

A window component that receives a click event. Created with classes in the ttk module.

button1 = ttk.Button(frame, text="Click me")

1. Button()

The Button() constructor creates button objects, and takes two arguments, the parent component and the text that the button displays.

button1 = ttk.Button(frame, text="Click me")

1. columnspan [argument]

This attribute of the grid method indicates how many columns a component should span.

This will create a component that spans columns 1, 2, and 3 in the first row.

widget.grid(row=0, column=1, columnspan=3)

1. command [argument]

An argument to the Button() constructor that specifies the function that is to be executed when clicked. This is called a callback function or event handler.

This calls a function named click\_button1()

button1 = ttk.Button(frame, text="Click me", command=click\_button1)

1. component

A component is one element in a GUI, and can be things such as labels, entry fields, and buttons. These components can also be called widgets.

1. destroy()

This method of the root windows that causes the event processing loop to end, which ends the program.

This calls a function named click\_button2() that has a destroy command

button2 = ttk.Button(frame, text="Click me", command=click\_button2)

def click\_button2():

root.destroy()

1. entry field

another name for a text entry field

1. Entry()

A constructor that creates an entry with the width in characters. It takes parameters for parent, width, and textvariable (used to bind a StringVar to the entry object.

investmentText = tk.StringVar # creates the StringVar object

# binds the StringVar to the text entry field

investmentEntry = ttk.Entry(frame, width=25, textvariable=investmenText

1. event callback function

this is the function that contains the code that should be executed when the event occurs, such as a button being clicked.

def click\_button1():

root.title("You clicked the button")

def click\_button2():

root.destroy()

1. event handler

another name for "event callback function"

1. event processing loop

This is a state in which the program is listening for events that it needs to recognize and respond it, such as the user clicking on a button.

1. frame

A invisible container that use to group components together. Frames and buttons are created with classes in the ttk module.

[The winfo\_children() method gets all of the components in the frame. Then you can use a loop to appy padding to each component with grid\_configure()]

Create a frame and add it to the root window:

frame = ttk.Frame(root, padding="10 10 10 10")

frame.pack(fill=tk.BOTH, expand=True)

1. Frame()

The Frame() constructor in the ttk module creates frame objects. It takes two arguments, the parent component (what holds the frame), and the padding between the frame and the edge of the component. The second is optional but necessary.

frame = ttk.Frame(root, padding="10 10 10 10")

You can create a class that defines a frame: inherit from Frame superclass, then create subclass that has specific setup.

class InvestmentFrame(ttk.Frame):

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

ttk.Fram.))init\_\_(self, parent, padding="10 10 10 10")

(and so on)

1. geometry()

A method that accepts a string "width x height" to set the window size.

root.geometry("300x150")

1. get()

This method gets the string in a StringVar object

investment = investmentText.get()

1. graphical user interface (GUI)

A GUI is an interface to a program that uses visual elements to display information and receive user input. It includes components such as windows, buttons, labels, and text entry fields.

1. grid()

alternative to the pack() method. Lines up components in a grid rather than working one component at a time. Uses a structure of columns and row. Many arguments:

column - where component is added, index starts at 0

row - ditto

sticky - which side of the container the component should stick to

N W E S indicate which side of the box

padx - horizontal padding in pixels (default=0)

pady - vertical padding in pixels (default=0)

columnspan - the number of columns that the component should span

rowspan - the number of rows that the component should span

Build the label and entry objects then assign them to the grid:

ttk.Label(frame, text="Yearly Interest Rate:").grid(column=0, row=1, sticky=tk.E)

ttk.Entry(frame, width=25, textvariable=rateText).grid(column=1, row=1)

[The winfo\_children() method gets all of the components in the frame. Then you can use a loop to apply padding to each component with grid\_configure().]

1. label

labels label other components to show what they do. In a dialogue box, a label will tell the user what to enter in that text entry field.

1. Label()

constructor from the ttk module that creates a label. You can just display the label, and you do not need to assign it to a variable unless you need to change it.

this creates a label and displays it

investmentLabel = ttk.Label(frame, text="Monthly Investment")

invesmentLabel.pack()

1. mainloop()

This method is part of root window that is called just after the window is set up, and causes the program to enter an "event processing loop," that listens for events such as user input and mouse clicks. It makes the window visible and is essential.

Process of creating an empty root window:

root = tk.Tk()

root.title("Future Value")

root.geometry("300x150")

root.mainloop()

1. message box

a component created by the message module of the tkinter module.

Methods of the messagebox module. Each has two arguments (title, message)

showinfo

showwarning

showerror

askyesno - displays a message box with a question mark in a blue circle

askokcancel

from tkinter import messagebox

messagebox.showinfo("Info", "This application monitors keystrokes.")

response = messagebox.asktocancel("Confirm deletion")

if response == True:

# code that performs deletion

1. pack()

A method of the frame object that sizes the frame to be just large enough for the components in it, with two optional arguments, fill and expand.

frame.pack(fill=tk.BOTH, expand=True)

button1.pack()

1. root window

A root window is the basic starting point and foundation for a GUI interface. It is created and set up first when designing a GUI. Methods associated with a root window are title(), geometry(), and mainloop().

1. set()

Sets the string in a StringVar object

fvText.set("$2,000)

sets a value to the fvText object

1. showwarning()

A method of the messagebox module that shows a warning text with an OK button.

messagebox.showwarning("Be warned!")

1. sticky [argument]

This argument of the grid() method tells which side of the box a component should "stick" to. Defined as N S E or W for cardinal directions.

This keeps the label on the right side of the component:

ttk.Label(frame, text="Yearly Interest Rate:").grid(column=0, row=1, sticky=tk.E)

1. StringVar()

An object that contains a string that can be bound to other objects such as a text entry box.

#This constructor creates a StringVar object

investmentText = tk.StringVar()

#This creates a readonly text entry field and binds it to a StringVar object

investmentText = ttk.Entry(frame, width=25, textvariable=fvText, state="readonly")

1. text box

another name for a text entry field

1. text entry field

a component that accepts text entry. It is created with the Entry() constructor, and can have optional readonly attribute

investmentText = tk.StringVar # creates the StringVar object

# binds the StringVar to the text entry field

investmentEntry = ttk.Entry(frame, width=25, textvariable=investmenText

1. text field

another name for a text entry field

1. title()

This method is part of a GUI window object and sets the title of the window, appears as a string on the title bar.

root.title("Future Value Calculator")

1. Tk()

This is a constructor that creates the root window object. It is imported with the tkinter module. After this, you create a frame and components that go into the root window.

import tkinter

root = tkinter.Tk()

1. tkinter

tkinter is the standard Python interface to the Tcl/Tk GUI toolkit. You can test it on your system with the python command with 'python -m tkinter'. It should be imported at the start of the application: import tkinkter.

A screenshot of a computer

Description automatically generated

1. toolkit

A set of tools that allow the programmer to extend the capability of python. The tkinter package is a graphics interface toolkit.

1. ttk

This module is imported from the tk module, and contains classes that create frames and buttons. This creates a button object:

from tkinter import ttk

button1 = ttk.Button(frame, text="Click me")

1. widget

This is another term for a GUI component.

(38 total assignment items)