NIET
Greater Noida
Autonomous Institute
Estd.2001
Get Future Ready

CodSoft
Code.Create.Succeed

Internship Presentation in Python

GitHub: https://github.com/AshishSahani0/CodSoft_Python_Task

Department of Computer Science and Engineering

NIET Greater Noida
Autonomous Institute
Estd.2001
Get Future Ready

Presented By:
Ashish Sahani
CSE-B
2301330100058

Presented To:
Mr. Rupendra Kaushik
Ms. Ankita Sharma

NIET
Greater Noida
Autonomous Institute
Estd.2001
Get Future Ready

About CodSoft.....

It's aim is to help students lacking basic skills by offering hands-on learning through live projects and real-world examples through Internships in different background i.e. Python , Java, C , C++ Programming Language as well it also provide internship in different developing backgrounds such as Web Development, Android Development , Data Science etc.....

https://www.codsoft.in/

contact@codsoft.in

CodSoft
Code.Create.Succeed

INTERNSHIP OFFER LETTER

Date : **08/07/2024** **ID:CS11WX324372**

Dear,

Ashish Sahani

We would like to congratulate you on being selected for the " **Python Programming"** virtual internship position with **"CodSoft"**. We at **CodSoft** are excited that you will join our team.

The duration of the internship will be of **4 weeks**, starting from **10 July 2024 to 10 August 2024.** The internship is an educational opportunity for you hence the primary focus is on learning and developing new skills and gaining hands-on knowledge. We believe that you will perform all your tasks/projects.

As an intern, we expect you to perform all assigned tasks to the best of your ability and follow any lawful and reasonable instructions provided to you.

We are confident that this internship will be a valuable experience for you, we look forward to working with you and helping you achieve your career goals.

By accepting this offer, you commit to executing assigned tasks diligently and ensuring excellence in all aspects of your work.

Best of Luck!

Thank You!

Founder (CodSoft)

MSME Registered

www.codsoft.in
contact@codsoft.in

C.ID: a1abb96

CodSoft

CERTIFICATE

OF COMPLETION
PROUDLY PRESENTED TO

Ashish Sahani

has successfully completed 4 weeks of a virtual internship program in

Python Programming

with wonderful remarks at **CODSOFT** from **10/07/2024 to 10/08/2024.**

We were truly amazed by his/her showcased skills and invaluable contributions to the tasks and projects throughout the internship.

Founder

contact@codsoft.in www.codsoft.in Date: **13/08/2024**

Task 1…….. TO-DO LIST

NIET
Greater Noida
Autonomous Institute
Estd.2001
Get Future Ready

A To-Do List application is a useful project that helps users manage and organize their tasks efficiently. This project aims to create a command-line or GUI-based application using Python, allowing users to create, update, and track their to-do lists

```
4   class ToDoList():
5       def __init__(self):
6           self.tasks=[]
7           
8       def display_tasks(self):
9           print("To-Do-List")
10          for i, task in enumerate(self.tasks,1):
11              print(f"{i}. {task}")
12
13      def add_task(self):
14          self.task = input("Enter the  task: ")
15          if self.task != "":
16              self.tasks.append(self.task)
17              print(f"Task {self.task} is added successfully to the List.")
18
19      def delete_task(self):
20          task_number = int(input("Enter the task number to delete: "))
21          try:
22              del self.tasks[task_number - 1]
23              print(f"Task {task_number} is deleterd successfully.")
24          except IndexError:
25              print("Invalid task number.")
```

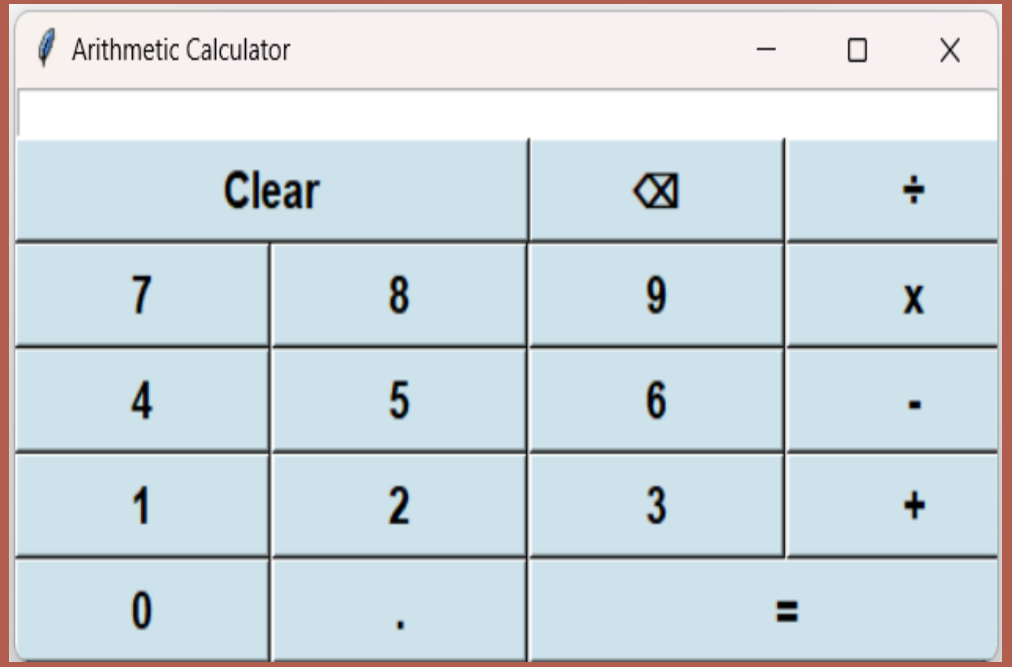
```
27      def update_task(self):
28          task_number = int(input("Enter the task number to be updated: "))
29          try:
30              task = input("Enter the new task: ")
31              if task != "":
32                  self.tasks[task_number - 1] = task
33                  print(f"Task {task_number} updated successfully.")
34          except IndexError:
35              print("Invalid task number.")
36
37  ToDo_List = ToDoList()
38  while True:
39      print("\nTo-Do-List Menu: ")
40      print("1. Display Tasks")
41      print("2. Add task")
42      print("3. Delete Task")
43      print("4. Update Task")
44      print("5. Quite")
45
```

```
46          choice = int(input("Enter the choice: "))
47
48          if choice == 1:
49              ToDo_List.display_tasks()
50          elif choice == 2:
51              ToDo_List.add_task()
52          elif choice == 3:
53              ToDo_List.delete_task()
54          elif choice == 4:
55              ToDo_List.update_task()
56          elif choice == 5:
57              print("Goodbye!")
58              break
59          else:
60              print("Invalid choice. Please Enter correct number: ")
61
62
```

NIET
Greater Noida
Autonomous Institute
Estd.2001
Get Future Ready

Task 2.... CALCULATOR

Design a simple calculator with basic arithmetic operations using GUI.



```python
from tkinter import *

window = Tk()
window.title("Arithmetic Calculator")
window.geometry("500x230")
window.config(bg="#F0FFFF")

def button_clicked():
    entry1.delete(0,END)

def button_backspace_clicked():
    entry1.delete(len(entry1.get())-1, END)

def button_number_clicked(number):
    current_text = entry1.get()
    entry1.delete(0, END)
    entry1.insert(0, current_text + str(number))

def button_operator(operator):
    current_text = entry1.get()
    entry1.delete(0,END)
    entry1.insert(0, current_text + str(operator))

def button_equals_clicked():
    try:
        result = eval(entry1.get())
        entry1.delete(0, END)
        entry1.insert(0, result)
    except:
        entry1.delete(0, END)
        entry1.insert(0, "Error")



frame = Frame(window)
frame.pack()
```

```
37
38      entry1 = Entry(frame,font=("Arial",10))
39      entry1.config(width=74)
40      entry1.grid(row=0, column=0, columnspan=4)
41
42      button_clear = Button(frame, text="Clear", font=("Arial Bold", 15),command=button_clicked)
43      button_clear.config(width=21,bg="#CDE2EA")
44      button_clear.grid(row=1, column=0, columnspan=2)
45
46      button_backspace = Button(frame, text="⌫", font=("Arial Bold", 15),command=button_backspace_clicked)
47      button_backspace.config(width=10 ,bg="#CDE2EA")
48      button_backspace.grid(row=1, column=2)
49
50      button_divide = Button(frame, text="÷", font=("Arial Bold", 15),command=lambda:button_operator("/"))
51      button_divide.config(width=10 ,bg="#CDE2EA")
52      button_divide.grid(row=1, column=3)
53
54      button7 = Button(frame, text="7", font=("Arial Bold", 15),command=lambda:button_number_clicked(7))
55      button7.config(width=10 ,bg="#CDE2EA")
56      button7.grid(row=2, column=0)
57
58      button8 = Button(frame, text="8", font=("Arial Bold", 15),command=lambda: button_number_clicked(8))
59      button8.config(width=10 ,bg="#CDE2EA")
60      button8.grid(row=2, column=1)
61
62      button9 = Button(frame, text="9", font=("Arial Bold", 15),command=lambda:button_number_clicked(9))
63      button9.config(width=10 ,bg="#CDE2EA")
64      button9.grid(row=2, column=2)
65
```

```
65
66      button_multiply = Button(frame, text="x", font=("Arial Bold", 15),command=lambda:button_operator("*"))
67      button_multiply.config(width=10 ,bg="#CDE2EA")
68      button_multiply.grid(row=2, column=3)
69
70      button4 = Button(frame, text="4", font=("Arial Bold", 15),command=lambda:button_number_clicked(4))
71      button4.config(width=10 ,bg="#CDE2EA")
72      button4.grid(row=3, column=0)
73
74      button5 = Button(frame, text="5", font=("Arial Bold", 15),command=lambda:button_number_clicked(5))
75      button5.config(width=10 ,bg="#CDE2EA")
76      button5.grid(row=3, column=1)
77
78      button6 = Button(frame, text="6", font=("Arial Bold", 15),command=lambda:button_number_clicked(6))
79      button6.config(width=10 ,bg="#CDE2EA")
80      button6.grid(row=3, column=2)
81
82      button_subtract = Button(frame, text="-", font=("Arial Bold", 15),command=lambda:button_operator("-"))
83      button_subtract.config(width=10 ,bg="#CDE2EA")
84      button_subtract.grid(row=3, column=3)
85
86      button1 = Button(frame, text="1", font=("Arial Bold", 15),command=lambda:button_number_clicked(1))
87      button1.config(width=10 ,bg="#CDE2EA")
88      button1.grid(row=4, column=0)
89
90      button2 = Button(frame, text="2", font=("Arial Bold", 15),command=lambda:button_number_clicked(2))
91      button2.config(width=10 ,bg="#CDE2EA")
92      button2.grid(row=4, column=1)
```

```
    button3 = Button(frame, text="3", font=("Arial Bold", 15),command=lambda:button_number_clicked(3))
    button3.config(width=10 ,bg="#CDE2EA")
    button3.grid(row=4, column=2)

    button_add = Button(frame, text="+", font=("Arial Bold", 15),command=lambda:button_operator("+"))
    button_add.config(width=10 ,bg="#CDE2EA")
    button_add.grid(row=4, column=3)

    button0 = Button(frame, text="0", font=("Arial Bold", 15),command=lambda:button_number_clicked(0))
    button0.config(width=10 ,bg="#CDE2EA")
    button0.grid(row=5, column=0)

    button_decimal = Button(frame, text=".", font=("Arial Bold", 15),command=lambda:button_operator("."))
    button_decimal.config(width=10 ,bg="#CDE2EA")
    button_decimal.grid(row=5, column=1)

    button_equals = Button(frame, text="=", font=("Arial Bold", 15),command=button_equals_clicked)
    button_equals.config(width=21 ,bg="#CDE2EA")
    button_equals.grid(row=5, column=2, columnspan=2)

    window.mainloop()
```

Task 3..... Rock-Paper-Scissors Game

User Input: Prompt the user to choose rock, paper, or scissors.

Computer Selection: Generate a random choice (rock, paper, or scissors) for the computer.

Game Logic: Determine the winner based on the user's choice and the computer's choice. Rock beats scissors, scissors beat paper, and paper beats rock.

Display Result: Show the user's choice and the computer's choice. Display the result, whether the user wins, loses, or it's a tie.

Score Tracking (Optional): Keep track of the user's and computer's scores for multiple rounds.

Play Again: Ask the user if they want to play another round.

User Interface: Design a user-friendly interface with clear instructions and feedback.

```python
import random
choices = ["Rock","Paper","Scissor"]
print("Choices:")
print("1. Rock")
print("2. Paper")
print("3. Scissor")
print("4. Quit")
user_wincount = 0
computer_wincount = 0
tie = 0
while True:
    user_choice = input("Enter your choice: ")
    if user_choice == "Quit":
        print("GoodBye!")
        break

    computer_choice = random.choice(choices)
    print(f"\nUser choice: {user_choice} \nComputer Choice: {computer_choice}")

```

```
21          if user_choice == computer_choice:
22              tie = tie + 1
23              print("It's a tie")
24          elif (user_choice == "Rock" and computer_choice == "Scissor") or \
25               (user_choice == "Paper" and computer_choice == "Rock") or \
26               (user_choice == "Scissor" and computer_choice == "Paper"):
27              user_wincount = user_wincount + 1
28              print("You win!")
29          else:
30              computer_wincount = computer_wincount + 1
31              print("Computer wins!")
32
33      print(f"User Win: {user_wincount}")
34      print(f"Computer Win: {computer_wincount}")
35      print(f"Tie : {tie}")
```

THANK YOU