PROGRAM 3: OPPS CONCEPT IN PYTHON

Requirement:

Dr. Vasi, a brilliant scientist with the help of Robotics,AI and ML has built a super robot Chitti,with speed 1 THz,Memory 1TB and capable of recognizing humman emotions.Smart Chitti now making his duplicates. Dr.Vasi is afr iad whether he will use the replicas for constructive or destructive purpose. Clarify his doubt by iplementing a program that involves hybrid inheritance to showcase the thought process of Chitti behind his own replicas.

Note: Highlight various OOPS concept such as inheritance,dtat abstraction,polymorphism,etc,in this conceptual program.

In [2]:
```
#importing libraries

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from multipledispatch import dispatch
from googletrans import Translator
translator = Translator()
```

In [3]:
```
# validation for robot type whether constructive or destructive
def valid_type_input():
    input1=["Constructive","CONSTRUCTIVE","constructive","Destructive","destructive","DESTRUCTIVE"]
    c=True
    while(c):
        n=input("Enter the value:")
        if n not in input1:
            print("Input not valid")
            c=True
        else:
            c=False
    return n
```

In [4]:
```
# validation for input of language
def valid_lang_input():
    language=["Hindi","Urdu","Punjabi" ,"Marathi","Telugu","Tamil" ,"Gujarati" ,"Kannada" ,"Malayalam","English"]
    c=True
    while(c):
        n=input("Enter the value:")
        if n not in language:
            print("Input not valid")
            c=True
        else:
            c=False
    return n
```

In [5]:
```
# validation for selecting con option
def valid_con_number():
    numbers={"1":"Manufacturing Robot","2":"Housekeeping Robot","3":"Medical Robot"}
    x=True
    while(x):
        n=input("Enter the value:")
        if n.isnumeric():
            if n not in numbers.keys():
                print("Input not valid")
                x=True
            else:
                x=False

    return numbers[n]
```

In [6]:
```
# validation for selecting des option
def valid_des_number():
    numbers={"1":"Military Robot","2":"Bank Robber Robot","3":"Terrorist Robot"}
    x=True
    while(x):
        n=input("Enter the value:")
        if n.isnumeric():
            if n not in numbers.keys():
                print("Input not valid")
                x=True
            else:
                x=False

    return numbers[n]
```

In [7]:
```
# validation for emotion scores
def valid_score():
    k=["1","2","3","4","5"]
    x=True
    while(x):
        n=input("Enter the value:")
        if n not in k:
            print('Invalid Input')
            x=True

        else:
            x=False
            return int(n)
```

In [8]:
```
# Specifications of robot
class Specification_display:
    def __init__(self):
        self.speed = "200"
        self.CPU = "EBX - 800 MHz"
        self.SystemMemory = "1 TB"
        self.Chipset = "82C868"
    def Specification_view(self):
        print("\t  Speed of the robot : ",self.speed)
        print("\t  Memory capacity    : ",self.SystemMemory)
```

In [9]:
```
# Robot recognizing person's emotions
class Person_Emotions:
    def __init__(self):
        self.Conpictures = ["C:/Users/Admin/Desktop/1MDS/Extra/resume & pic/Robot3.png","C:/Users/Admin/Desktop/1MDS/Ext
        self.Despictures = ["C:/Users/Admin/Desktop/1MDS/Extra/resume & pic/Robot4.jpeg","C:/Users/Admin/Desktop/1MDS/Ex

        self.Emo = ["Happiness 😊","Sadness ☹️","Surprise 😳"]
        self.Emo1 = ["Anger 😠","Disgust 😖","Fear 😨"]

        print("-------------------------------------------------------------------------------------------------------"
        print()
        print("👩SUJI   : Chitti can you recognise the emotion of Dr.Vasi? ")
        print("          : So that I'll know whether he is benifited out of this program or not.")
        print()
        print("🤖CHITTI : Sure Ma'am")
        print("          : Dr Vasi, I'll ask you 6 questions and you have to answer them by rating yourself from 1(rarel
        print("\t\t\t 1- ☆")
        print("\t\t\t 2- ☆☆")
        print("\t\t\t 3- ☆☆☆")
        print("\t\t\t 4- ☆☆☆☆")
        print("\t\t\t 5- ☆☆☆☆☆")
        print()
        print("👨Dr.VASI : Sure Chitti!")
        print("\n\t\t{*****  The questions below are asked by Chitti to Dr.Vasi *****}")
        print()
        print("-------------------------------------------------------------------------------------------------------"
        print()
        print("How often you feeling upset ? : ")
        q1 = valid_score()
        print("How often you have mood swings ? : ")
        q2 = valid_score()
        print("How often you laugh louder ? : ")
        q3 = valid_score()
        self.sum = q1+q2+q3

        print("How frequent you interact with other people ? ")
        q4 = valid_score()
        print("How concern you're in taking risk in life ? ")
        q5 = valid_score()
        print("How you feel that your life wasn't worthwhile ? ")
        q6 = valid_score()
        self.sum1 = q4+q5+q6

    def Emo_check(self):
        if (self.sum > self.sum1):
            if self.sum > 0 and self.sum <= 5:
                self.person = self.Emo[1]
            elif self.sum > 5 and self.sum <= 10:
                self.person = self.Emo[2]
            elif self.sum > 10 and self.sum <= 15:
                self.person = self.Emo[0]
        elif (self.sum < self.sum1):
            if self.sum1 > 0 and self.sum1 <= 5:
                self.person = self.Emo1[2]
            elif self.sum1 > 5 and self.sum1 <= 10:
                self.person = self.Emo1[1]
            elif self.sum1 > 10 and self.sum1 <= 15:
                self.person = self.Emo1[0]
        elif (self.sum == self.sum1):
            self.person = "having Mixed Feelings"

        else:
            print("Please select from the given scale")

    def Emo_Display(self):
        Person_Emotions.Emo_check(self)
        print()
        print("-------------------------------------------------------------------------------------------------------"
        print("🤖CHITTI   : Dr. Vasi is "+ self.person)
        print("\n-------------------------------------------------------------------------------------------------------
        print()
        print("\t\t\t************ END OF THE PROGRAM ************")
        print("\n-------------------------------------------------------------------------------------------------------
        print()
        print("👨Dr VASI  : Thank You Suji, with the help of your program now I understood what Chitti tried to convey m
        print()
        print("👩SUJI     : Your Welcome Dr.Vasi")
        print()
        print("-------------------------------------------------------------------------------------------------------"
```

In [10]:
```
# Languages spoken by robot
class Language:

    def __init__(self):

        print()
        self.eng=["Hindi","Urdu" ,"Punjabi" ,"Marathi","Telugu","Tamil" ,"Gujarati" ,"Kannada" ,"Malayalam"]
        print("-----------------------------------------------------------------------------------------------------")
        print("🤖CHITTI : Which Language do you perfer Dr.Vasi ")
        print()
        for i in range(0,7):
            print(self.eng[i])
        print()
        print("Enter the languge in english:")
        self.i= valid_lang_input()

    def c_set_data(self):
        print("Different Robots:")
        print("================")
        self.cr=['1. Manufacturing Robot','2. Housekeeping Robot','3. Medical Robot']
        for i in range(0,3):
            print(self.cr[i])
        print("\n\nEnter the number of Robot you choose :")
        self.robot = valid_con_number()
        print()
        print("\nList of languages:")
        print("====================")
        for i in range(0,7):
            print(self.eng[i])
        print("English")
        print()
        print("Enter the language you want the duplicate robots to speak from the above list: ")
        self.language=valid_lang_input()

    def d_set_data(self):
        print("Different Robots:")
        print("================")
        self.dr=['1. Military Robot','2. Bank Robber Robot','3. Terrorist Robot']
        for i in range(0,3):
            print(self.dr[i])
        print("\n\nEnter the number of Robot you choose :")
        self.robot = valid_des_number()
        print()
        print("\nList of languages:")
        print("====================")
        for i in range(0,7):
            print(self.eng[i])
        print("English")
        print()
        print("Enter the language you want the duplicate robots to speak from the above list: ")
        self.language=valid_lang_input()


    @dispatch(str)
    def robot_lang(self,i):
        self.i=i
        self.engl={"Hindi":"hi","Urdu":"ur","Punjabi":"pa","Marathi":"mr","Telugu":"te","Tamil":"ta","Gujarati":"gu","Ka
        print("-----------------------------------------------------------------------------------------------------")
        if self.i in self.engl:
            r=translator.translate('Nice to meet you', src='en', dest=self.engl[self.i])
            print("🤖CHITTI: {}".format(r.text))

        #for i in :
        #    r=translator.translate('Nice to meet you', src='en', dest=i)
        #    print("🤖CHITTI: {}".format(r))

        #r1 = translator.translate('Nice to meet you', src='en', dest='hi')
        #r2 = translator.translate('Nice to meet you', src='en', dest='ur')
        #r3 = translator.translate('Nice to meet you', src='en', dest='pa')
        #r4 = translator.translate('Nice to meet you', src='en', dest='mr')
        #r5 = translator.translate('Nice to meet you', src='en', dest='te')
        #r6 = translator.translate('Nice to meet you', src='en', dest='ta')
        #r7 = translator.translate('Nice to meet you', src='en', dest='gu')
        #r8 = translator.translate('Nice to meet you', src='en', dest='kn')
        #r9 = translator.translate('Nice to meet you', src='en', dest='ml')
        #self.lang =[r1,r2,r3,r3,r4,r5,r6,r7,r8,r9]
        #self.lang = ["🤖CHITTI: आपसे मिलकर खुशी हुई","🤖CHITTI: آپ سے مل کر خوشی ہوئی", "🤖CHITTI: ਤੁਹਾਨੂੰ ਮਿਲ ਕੇ ਖੁਸ਼ੀ ਹੋ

        #if self.i in self.eng:
        #    k=self.eng.index(self.i)
        #    print(self.lang[k])

    @dispatch(str,str)
    def robot_lang(self,robot,language):
        self.robot=robot
        self.language=language
        print("The duplicate robots created will be used for following purpose with specified language:")
        print("\n\t\tRobot purpose type : "+ self.robot)
```

```
        print("\t\tLanguage           : " + self.language)

    def display_speech(self):
        self.robot_lang(self.i)
        print()

    def display_speech1(self):
        self.robot_lang(self.robot,self.language)
```

In [11]:
```
# Constructive type of robot
class Constructive(Specification_display):
    def __init__(self):
        self.Conpictures = ["C:/Users/Admin/Desktop/pictures/robot/CONS.jpeg"]
        self.func=["Laboratory","Medical","Scientifical research"," Housekeeping","Manufacturing"]

    def con_display(self):
        fig=plt.figure(figsize=(8,8))
        pic = self.Conpictures[0]
        img=mpimg.imread(pic)
        plt.imshow(img)
        plt.axis('off')
        plt.show("\n")
        Specification_display.__init__(self)
        Specification_display.Specification_view(self)
        print()
        print()
        d='Functions of Constructive'
        print(""+"*"+'-'*40+"*")
        print(""+"|{:^40s}|".format(d.upper()))
        print(""+"*"+'-'*40+"*")

        for i in range(0,4):
            print(""+"|{:^40s}|".format(self.func[i]))
            #print(self.func[i])
        print(""+"*"+'-'*40+"*")
```

In [12]:
```
# Destructive type of robot
class Destructive(Specification_display):
    def __init__(self):
        self.Despictures = ["C:/Users/Admin/Desktop/pictures/robot/Des.jpeg"]
        self.fun=["Soldier","Terrorist","Bank Robbery","Crime"]

    def des_display(self):
        fig=plt.figure(figsize=(8,8))
        pic = self.Despictures[0]
        img=mpimg.imread(pic)
        plt.imshow(img)
        plt.axis('off')
        plt.show("\n")
        Specification_display.__init__(self)
        Specification_display.Specification_view(self)
        print()
        print()
        s='Functions of Destructive'
        print(""+"*"+'-'*40+"*")
        print(""+"|{:^40s}|".format(s.upper()))
        print(""+"*"+'-'*40+"*")

        for i in range(0,4):
            print(""+"|{:^40s}|".format(self.fun[i]))
            #print(self.fun[i])
        print(""+"*"+'-'*40+"*")
```

In [13]:
```
# Chitti the robot
class Chitti(Constructive,Destructive,Person_Emotions,Language):

    def view(self):
        if(self.type == "Constructive"):
            print()
            Language.__init__(self)
            print()
            Language.display_speech(self)
            print()
            Constructive.__init__(self)
            print()
            Constructive.con_display(self)
            print()
            Language.c_set_data(self)
            print()
            Language.display_speech1(self)
            print()
            Person_Emotions.__init__(self)
            print()
            Person_Emotions.Emo_Display(self)

        elif(self.type=="Destructive"):
            print()
            Language.__init__(self)
            print()
            Language.display_speech(self)
            print()
            Destructive.__init__(self)
            print()
            Specification_display.__init__(self)
            print()
            Specification_display.Specification_view(self)
            print()
            Destructive.des_display(self)
            print()
            Language.d_set_data(self)
            print()
            Language.display_speech1(self)
            print()
            Person_Emotions.__init__(self)
            print()
            Person_Emotions.Emo_Display(self)


    def __init__(self):
        self.pictures = ["C:/Users/Admin/Desktop/pictures/robot/Main.jpg"]
        fig=plt.figure(figsize=(15,25))
        pic = self.pictures[0]
        img=mpimg.imread(pic)
        plt.imshow(img)
        plt.axis('off')
        plt.show("\n")
        print()
        print("-------------------------------------------------------------------------------------------------------")
        print("\t\t{ ***** After the conversation Between Suji and Chitti ****}")
        print("\t{ ***** She has come up with a program to help Dr.Vasi in Better understanding  ****}")
        print()
        print("👩SUJI     : Hi Dr Vasi! Can we jump into the program for better understanding?")
        print()
        print("👨Dr VASI  : Sure Suji 😊 ")
        print("-------------------------------------------------------------------------------------------------------")
        print("\n\n\t\t\t\t PROGRAM : VASI AND HIS CONFUSION????")
        print("\t\t\t\t **********************************")
        print("\nEnter the purpose [Constructive or Destructive] :")
        self.type = valid_type_input()
        self.view()
```

In [15]:
```
s=Chitti()
```

Dr. Vasi, a brilliant scientist.
He was recently afraid whether to use the replicas for constructive or destructive purpose, and decided to get help from his friend Suji who is a programmer.

Hello Suji,

My invention Chitti the Robot is now making his duplicates and I'm more concern about the purpose whether it is used for constructive or destructive. It will be great if you could use your programming skills and explain on what Chitti trying to convey.

Hi Dr,

I'm glad, working for a scientist who worked on Robotics, AI, ML to built a super humanoid robot. I'll code a program using the oops concept for better understanding along with the insights and information given by Chitti.

{ ***** After the conversation Between Suji and Chitti ****}
{ ***** She has come up with a program to help Dr.Vasi in Better understanding ****}

👩SUJI : Hi Dr Vasi! Can we jump into the program for better understanding?

🧑Dr VASI : Sure Suji 😊

PROGRAM : VASI AND HIS CONFUSION????
****

```
Enter the purpose [Constructive or Destructive] :
Enter the value:Destructive
```

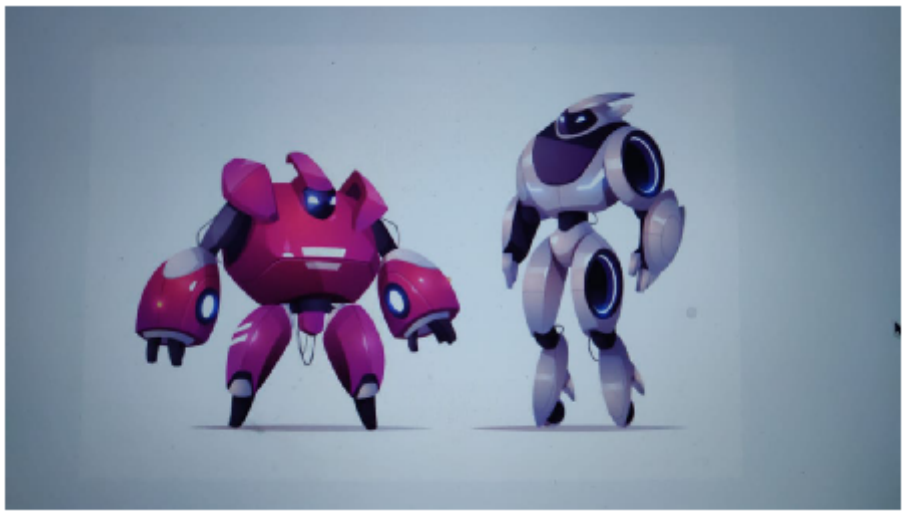
🤠CHITTI : Which Language do you perfer Dr.Vasi

```
Hindi
Urdu
Punjabi
Marathi
Telugu
Tamil
Gujarati
```

Enter the languge in english:
Enter the value:Tamil

🤖CHITTI: உங்களை சந்திப்பதில் மகிழ்ச்சி

Speed of the robot : 200
Memory capacity   : 1 TB



Speed of the robot : 200
Memory capacity   : 1 TB

```
*----------------------------------------*
|        FUNCTIONS OF DESTRUCTIVE        |
*----------------------------------------*
|                Soldier                 |
|               Terrorist                |
|              Bank Robbery              |
|                 Crime                  |
*----------------------------------------*
```

Different Robots:
==========
1. Military Robot
2. Bank Robber Robot
3. Terrorist Robot

Enter the number of Robot you choose :
Enter the value:1

List of languages:
==========
Hindi
Urdu
Punjabi
Marathi
Telugu
Tamil
Gujarati
English

Enter the language you want the duplicate robots to speak from the above list:
Enter the value:English

The duplicate robots created will be used for following purpose with specified language:

Robot purpose type : Military Robot
Language   : English

👩SUJI   : Chitti can you recognise the emotion of Dr.Vasi?
         : So that I'll know whether he is benifited out of this program or not.

🤖CHITTI : Sure Ma'am
         : Dr Vasi, I'll ask you 6 questions and you have to answer them by rating yourself from 1(rarely) to 5(always)

1- ☆
2- ☆☆
3- ☆☆☆
4- ☆☆☆☆
5- ☆☆☆☆☆

👦Dr.VASI : Sure Chitti!

{***** The questions below are asked by Chitti to Dr.Vasi *****}

How often you feeling upset ? :
Enter the value:4
How often you have mood swings ? :
Enter the value:3
How often you laugh louder ? :
Enter the value:4
How frequent you interact with other people ?
Enter the value:5
How concern you’re in taking risk in life ?
Enter the value:4
How you feel that your life wasn’t worthwhile ?
Enter the value:3

👿CHITTI   : Dr. Vasi is Anger 😠

********** END OF THE PROGRAM **********

👨Dr VASI : Thank You Suji, with the help of your program now I understood what Chitti tried to convey me
👩SUJI   : Your Welcome Dr.Vasi
