Polymorphism is the concept that a single entity can represent many different types within a program. It allows programmers to define methods within the child class while having the same name as what is defined in the parent class. Essentially, the same function can be used for different types. In Python you can use different class methods, functions, or objects to define polymorphism. Below are some examples of polymorphism and how it is used within python.

**Example 1: Basic Polymorphism**

Below are some basic examples of polymorphism. The first example uses integers, and the second example uses strings.

#polymorphism with numbers

number1 = 1

number2 = 4

print(number1 + number2)

#polymorphism with strings

string1 = "Hello"

string2 = "There"

print(string1 + " " + string2)

**Example 2: Polymorphism Within Classes**

Below is an example of polymorphism within a class. It takes two different types of objects, cars and trains, and uses the same variable names to describe them.

class Car:

    def \_\_init\_\_(item, name, color):

        item.name = name

        item.color = color

    def type(item):

        print(f"I am a {item.color} {item.name}")

    def speed(item):

        print("Since I am a car I can travel between 0 - 200 MPH")

class Train:

    def \_\_init\_\_(item, name, color):

        item.name = name

        item.color = color

    def type(item):

        print(f"I am a {item.color} {item.name}")

    def speed(item):

       print("Since I am a train I can travel between 0 - 500 MPH")

car1 = Car("Corvette", "Black")

car2 = Car("Ferrari", "Red")

train1 = Train("Steam Engine", "Yellow")

for vehicle in (car1, car2, train1):

    vehicle.type()

    vehicle.speed()

**Example Outputs:**

Below are what your outputs should look like for your example code.

Example 1:

Graphical user interface, text

Description automatically generated

Example 2:

Text

Description automatically generated

**Do It Yourself:**

Now that you have tried these examples, create a class using polymorphism which compares two countries, the United States and Germany. Define their capitals, primary language, and GDP.

**Do It Yourself Solution:**

#This creates the United States Class

class unitedStates:

    #defines the variables for capital, language, and gdp

    def \_\_init\_\_(item, capital, language, GDP):

        item.capital = capital

        item.language = language

        item.GDP = GDP

    #defines the capital function

    def cap(item):

        print(f"The United States' capital is {item.capital}")

    #defines the language function

    def lang(item):

        print(f"The United States' primary language is {item.language}")

    #defines the gdp function

    def gdp(item):

        print(f"The United States' GDP is {item.GDP}")

#This creates the Germany Class

class germany:

    #defines the variables for capital, language, and gdp

    def \_\_init\_\_(item, capital, language, GDP):

        item.capital = capital

        item.language = language

        item.GDP = GDP

    #defines the capital function

    def cap(item):

        print(f"Germany's capital is {item.capital}")

    #defines the language function

    def lang(item):

        print(f"Germany's primary language is {item.language}")

     #defines the gdp function

    def gdp(item):

        print(f"Germany's GDP is {item.GDP}")

#sets the values for the United States class

USA = unitedStates("Washington D.C.", "English", "$20.94 Trillion")

#sets the values for the Germany class

GER = germany("Berlin", "German", "$3.806 Trillion")

#calls the functions for each class

for country in (USA, GER):

    country.cap()

    country.lang()

    country.gdp()

Sources:

<https://www.geeksforgeeks.org/polymorphism-in-python/>

<https://www.programiz.com/python-programming/polymorphism>