****

Class: FY-MCA Shift / Div : F2 / B Roll Number : 51124

Name: Sameer Kakade Assignment No:8 Date of Implementation:4/11/2023

****

1. Program for Creating Classes and Objects.

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def display(self):
        print(f"Name: {self.name}, Age: {self.age}")

person1 = Person("John Doe", 30)
person1.display()

# Output:
# Name: John Doe, Age: 30
```

2. Program to demonstrate the Constructor Method.

```
class Student:
```

```
    def __init__(self, name, roll):
        self.name = name
        self.roll = roll

    def display(self):
        print(f"Name: {self.name}, Roll: {self.roll}")

student1 = Student("Alice", 101)
student2 = Student("Bob", 102)

student1.display()
student2.display()

# Output:
# Name: Alice, Roll: 101
# Name: Bob, Roll: 102



# 3. Program for creating Classes with Multiple Objects.

class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return 3.14 * self.radius**2
```

```
circle1 = Circle(5)

circle2 = Circle(7)


print(f"Area of Circle 1: {circle1.area()}")

print(f"Area of Circle 2: {circle2.area()}")


# Output:

# Area of Circle 1: 78.5

# Area of Circle 2: 153.94




# 4. Program to demonstrate the Class Attributes versus Data Attributes.


class Car:

    car_type = "SUV"


    def __init__(self, model):

        self.model = model


car1 = Car("Toyota")

car2 = Car("Honda")


print(f"Car 1 - Model: {car1.model}, Type: {car1.car_type}")

print(f"Car 2 - Model: {car2.model}, Type: {car2.car_type}")


# Output:

# Car 1 - Model: Toyota, Type: SUV
```

```
# Car 2 - Model: Honda, Type: SUV



# 5. Program to demonstrate Encapsulation.

class BankAccount:
    def __init__(self):
        self.balance = 0

    def deposit(self, amount):
        self.balance += amount
        print(f"Deposited {amount} units. New balance: {self.balance} units")

    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
            print(f"Withdrawn {amount} units. New balance: {self.balance} units")
        else:
            print("Insufficient funds")

account = BankAccount()
account.deposit(1000)
account.withdraw(500)

# Output:
# Deposited 1000 units. New balance: 1000 units
# Withdrawn 500 units. New balance: 500 units
```

```
# 6. Program to demonstrate Inheritance.


class Animal:
    def sound(self):
        print("Some generic sound")


class Dog(Animal):
    def sound(self):
        print("Bark")


class Cat(Animal):
    def sound(self):
        print("Meow")


animal = Animal()
dog = Dog()
cat = Cat()


animal.sound()
dog.sound()
cat.sound()


# Output:
# Some generic sound
# Bark
```

```
# Meow



# 7. Program to demonstrate Polymorphism.


def make_sound(animal):
    animal.sound()


animal = Animal()
dog = Dog()
cat = Cat()


make_sound(animal)
make_sound(dog)
make_sound(cat)


# Output:
# Some generic sound
# Bark
# Meow



# 8. Program to demonstrate few methods of Mathematics module.


import math


number = 16
```

```
print(f"Square root of {number}: {math.sqrt(number)}")

print(f"Factorial of {number}: {math.factorial(number)}")

print(f"Logarithm base 2 of {number}: {math.log2(number)}")


# Output:

# Square root of 16: 4.0

# Factorial of 16: 20922789888000

# Logarithm base 2 of 16: 4.0



# 9. Program to get the Internet Access.


import urllib.request


try:

    urllib.request.urlopen("http://www.google.com")

    print("Internet Access: Allowed")

except Exception as e:

    print(f"Internet Access: Denied, {e}")


# Output:

# Internet Access: Allowed



# 10. Program to display the various Date Time formats.
```

```
from datetime import datetime


current_datetime = datetime.now()


print(f"Current date and time: {current_datetime}")

print(f"Current year: {current_datetime.year}")

print(f"Month of year: {current_datetime.strftime('%B')}")

print(f"Week number of the year: {current_datetime.strftime('%U')}")

print(f"Weekday of the week: {current_datetime.strftime('%A')}")

print(f"Day of year: {current_datetime.strftime('%j')}")

print(f"Day of the month: {current_datetime.strftime('%d')}")

print(f"Day of week: {current_datetime.strftime('%w')}")


# Output:

# Current date and time: 2023-10-09 14:23:43.036527

# Current year: 2023

# Month of year: October

# Week number of the year: 40

# Weekday of the week: Sunday

# Day of year: 282

# Day of the month: 09

# Day of week: 0



# 11. Program to determine whether a given year is a leap year.


def is_leap_year(year):
```

```
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False

year = 2024
leap_year = is_leap_year(year)
print(f"{year} is a leap year: {leap_year}")

# Output:
# 2024 is a leap year: True


# 12. Program to convert a string to datetime.

from datetime import datetime

date_string = "2023-10-09"
date_obj = datetime.strptime(date_string, "%Y-%m-%d")
print(f"Converted date: {date_obj}")

# Output:
# Converted date: 2023-10-09 00:00:00


# 13. Program to subtract five days from current.
```

```
from datetime import datetime, timedelta

current_date = datetime.now()
new_date = current_date - timedelta(days=5)
print(f"Current Date: {current_date}")
print(f"Date 5 days ago: {new_date}")

# Output:
# Current Date: 2023-10-09 14:23:43.144162
# Date 5 days ago: 2023-10-04 14:23:43.144162


# 14. Program to convert unix timestamp string to readable date.

import datetime

timestamp = 1672531200
date_obj = datetime.datetime.fromtimestamp(timestamp)
print(f"Readable date: {date_obj}")

# Output:
# Readable date: 2023-12-31 00:00:00


# 15. Program to print day and date of yesterday, today, tomorrow.

from datetime import datetime, timedelta
```

```
current_date = datetime.now()

yesterday = current_date - timedelta(days=1)

tomorrow = current_date + timedelta(days=1)


print(f"Yesterday: {yesterday.strftime('%A, %Y-%m-%d')}")

print(f"Today: {current_date.strftime('%A, %Y-%m-%d')}")

print(f"Tomorrow: {tomorrow.strftime('%A, %Y-%m-%d')}")


# Output:

# Yesterday: Sunday, 2023-10-08

# Today: Monday, 2023-10-09

# Tomorrow: Tuesday, 2023-10-10"
```