

Python Programming - Lab - 11

March 11, 2025

Python Programming - 2301CS404

Lab - 11

OM BHUT | 23010101033 | 122

1 Modules

1.0.1 01) WAP to create Calculator module which defines functions like add, sub,mul and div.

1.0.2 Create another .py file that uses the functions available in Calculator module.

[]:

1.0.3 02) WAP to pick a random character from a given String.

[2]: `import random`

`s = "abcdef"`

`print(random.choice(s))`

f

1.0.4 03) WAP to pick a random element from a given list.

[4]: `import random`

`s = [1,34,56,7,4]`

`print(random.choice(s))`

34

1.1 04) WAP to roll a dice in such a way that every time you get the same number.

```
[11]: import random

random.seed(1)
s = random.randint(1,6)
print(s)

random.seed(2)
s = random.randint(1,6)
print(s)

random.seed(1)
s = random.randint(0,6)
print(s)
```

2
1
1

1.1.1 05) WAP to generate 3 random integers between 100 and 999 which is divisible by 5.

```
[13]: import random

count = 1
while count<=3:
    n = random.randint(100,999)
    if n%5==0:
        print(n)
        count+=1
```

885
705
425

1.1.2 06) WAP to generate 100 random lottery tickets and pick two lucky tickets from it and announce them as Winner and Runner up respectively.

```
[5]: import random

t = random.sample(range(100,999),100)
ticket1 = random.choice(t)
t.remove(ticket1)
ticket2 = random.choice(t)
print(ticket1,ticket2,sep=" ")
```

873 544

1.1.3 07) WAP to print current date and time in Python.

```
[19]: from datetime import datetime
      val = datetime.now()
      print(val)
```

2025-02-19 21:22:36.199047

1.1.4 08) Subtract a week (7 days) from a given date in Python.

```
[26]: from datetime import datetime, timedelta
      val = datetime.now() - timedelta(days=7)
      print(val)
```

2025-02-12 21:27:42.663669

1.1.5 09) WAP to Calculate number of days between two given dates.

```
[28]: date1 = datetime(2025, 2, 19)
      date2 = datetime(2025, 2, 26)

      date_diff = date2 - date1
      print(date_diff.days)
```

7

1.1.6 10) WAP to Find the day of the week of a given date.(i.e. whether it is Sunday/monday/tuesday/etc.)

```
[33]: date1 = datetime.now()
      days_of_week = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
                      ↪ "Saturday", "Sunday"]

      dayOfWeek = days_of_week[date1.weekday()]
      print(dayOfWeek)
```

Wednesday

1.1.7 11) WAP to demonstrate the use of date time module.

```
[1]: import datetime

      # Get the current date and time
      now = datetime.datetime.now()
      print(f"Current Date and Time: {now}")

      # Get today's date
```

```

today = datetime.date.today()
print(f"Today's Date: {today}")

# Extract specific parts of the date and time
year = now.year
month = now.month
day = now.day
hour = now.hour
minute = now.minute
second = now.second
print(f"Year: {year}, Month: {month}, Day: {day}")
print(f"Hour: {hour}, Minute: {minute}, Second: {second}")

# Creating a specific date object
specific_date = datetime.date(2025, 2, 19)
print(f"Specific Date: {specific_date}")

# Add 10 days to the current date
future_date = today + datetime.timedelta(days=10)
print(f"10 days from Today: {future_date}")

# Subtract 5 days from the current date
past_date = today - datetime.timedelta(days=5)
print(f"5 days ago: {past_date}")

# Format a date into a string
formatted_date = now.strftime("%A, %B %d, %Y %I:%M%p")
print(f"Formatted Date: {formatted_date}")

```

Current Date and Time: 2025-02-19 21:46:57.961491
 Today's Date: 2025-02-19
 Year: 2025, Month: 2, Day: 19
 Hour: 21, Minute: 46, Second: 57
 Specific Date: 2025-02-19
 10 days from Today: 2025-03-01
 5 days ago: 2025-02-14
 Formatted Date: Wednesday, February 19, 2025 09:46PM

1.1.8 12) WAP to demonstrate the use of the math module.

```

[2]: import math

# 1. Getting the value of pi
pi_value = math.pi
print(f"Value of Pi: {pi_value}")

# 2. Square root of a number

```

```

num = 16
sqrt_value = math.sqrt(num)
print(f"Square root of {num}: {sqrt_value}")

# 3. Power of a number
base = 2
exponent = 3
power_value = math.pow(base, exponent)
print(f"{base} raised to the power of {exponent}: {power_value}")

# 4. Trigonometric functions
angle_deg = 45 # degrees
angle_rad = math.radians(angle_deg) # Convert degrees to radians
sin_value = math.sin(angle_rad)
cos_value = math.cos(angle_rad)
print(f"Sine of {angle_deg} degrees: {sin_value}")
print(f"Cosine of {angle_deg} degrees: {cos_value}")

# 5. Logarithmic functions
log_value = math.log(100, 10) # log base 10
print(f"Logarithm of 100 to the base 10: {log_value}")

# 6. Factorial of a number
fact_value = math.factorial(5)
print(f"Factorial of 5: {fact_value}")

# 7. Absolute value
neg_num = -7
abs_value = math.fabs(neg_num)
print(f"Absolute value of {neg_num}: {abs_value}")

# 8. Rounding a number
number = 4.567
rounded_value = round(number, 2)
print(f"{number} rounded to 2 decimal places: {rounded_value}")

# 9. Greatest common divisor (GCD)
gcd_value = math.gcd(36, 60)
print(f"GCD of 36 and 60: {gcd_value}")

# 10. Converting radians to degrees
radian_value = math.pi / 4
degree_value = math.degrees(radian_value)
print(f"{radian_value} radians is equal to {degree_value} degrees")

```

Value of Pi: 3.141592653589793

Square root of 16: 4.0

2 raised to the power of 3: 8.0

Sine of 45 degrees: 0.7071067811865476
Cosine of 45 degrees: 0.7071067811865476
Logarithm of 100 to the base 10: 2.0
Factorial of 5: 120
Absolute value of -7: 7.0
4.567 rounded to 2 decimal places: 4.57
GCD of 36 and 60: 12
0.7853981633974483 radians is equal to 45.0 degrees