

Set B

1.00 to 4pm

2/07/21

20MCA131

Programming Lab

~~ICEM~~

ICE20MCA2028

B.

1. Generate Fibonacci Series of N terms.

2. Create a class time with private attributes, hour, minute and second. Overload + operator to find sum of 2 time.

1. $n = \text{int}(\text{input}(\text{"Enter the number of terms: "}))$

$f_1, f_2 = 0, 1$

$f_3 = f_1 + f_2$

$\text{Print}(\text{"Fibonacci Series of first ", n, " terms is:"})$

$\text{Print}(f_1)$

$\text{Print}(f_2)$

$\text{for } i \text{ in range}(3, n+1)$

$\text{Print}(f_3)$

$f_1 = f_2$

$f_2 = f_3$

$f_3 = f_1 + f_2.$

Algorithm

Step 1:- Start

Step 2:- Declare variable f_1 & f_2 and initialise of 1 respectively.

Step 3:- Declare variable $f_3 = f_1 + f_2$

Step 4:- Print f_1 & f_2 .

Step 5:- Read the range & store in n , declare $i = 3$.

Step 6:- Repeat Step 7 to 12 until $i \leq n$.

Step 7:- print f_3

Step 8:- $f_1 = f_2$

Step 9:- $f_2 = f_3$

Step 10:- $f_3 = f_1 + f_2$

Step 11:- $i = i + 1$

Step 12:- Stop.

expected output

Enter the number of terms : 6

Fibonacci series of 6 terms is

0

1

1

2

3

5

tested output

Enter the number of terms : 7

Fibonacci series of 7 terms:

0

1

1

2

3

5

8

2. class time:

```
def init (self, h, m, s):
```

```
    self.hr = h
```

```
    self.min = m
```

```
    self.sec = s
```

```
def add (self, other):
```

```
    tempsec = self.sec + other.sec
```

```
    tempmin = self.min + other.min
```

```
    tempmin = tempsec / 60
```

```
    self.sec = int(tempsec % 60)
```

```
    self.min = self.min + other.min + tempmin
```

```
    temphr = self.min / 60
```

```
    self.min = int(self.min % 60)
```

```
    self.hr = int(self.hr + other.hr + temphr)
```

```
    return time(self.hr, self.min, self.sec)
```

```
def str(self):
```

```
    return str(self.hr) + 'hr' + str(self.min) +
```

```
    'min' + str(self.sec) + 'sec'
```

```
a = int(input("Enter hour of t1:"))
```

```
b = int(input("Enter min of t1:"))
```

```
c = int(input("Enter sec of t1:"))
```

```
x = int(input("Enter hour of t2:"))
```

```
y = int(input("Enter min of t2:"))
```

```
z = int(input("Enter sec of t2:"))
```

```
t1 = time(a, b, c)
```

```
t2 = time(x, y, z)
```

```
print(t1 + t2)
```

Algorithm

- Step 1: Start
- Step 2: Declare a class time
- Step 3: Define a method with attributes h, m & s.
initialise, self.h = h
self.m = m
self.s = s.
- Step 4: Define a method add with attribute other
Declare variable tempsec = self.s + other.s
Declare variable tempmin = tempsec / 60
~~next~~ self.s = integer of (tempsec % 60)
- Step 5: self.m = self.m + other.m + temp.m
temp.h = self.m / 60
self.m = integer of (self.m % 60)
self.h = integer of (self.h + other.h + temp.h)
- Step 6: return time (self.h + other.h + temp.h)
- Step 7: return string(self.h) + 'hr' + string of self.m +
'min' + string(self.s) + 'sec'.
- Step 8: Stop.

Main

- Step 1: Start
- Step 2: Read Value of hour of t₁, minute and
seconds of t₁.
- Step 3: Read Value of hour, minute and second of
t₂.
- Step 4: Call class time with attributes a, b, c of x, y, z.
into which it returned to variable t1, t2 respectively
- Step 5: print t₁ + t₂

Step 6: Stop

started output.

~~enter value of hour~~

enter hour of t_1 : 1

enter minutes of t_1 : 20

enter seconds of t_1 : 30

enter hour of t_2 : 1

enter minutes of t_2 : 20

enter seconds of t_2 : 15

2 hr 40 min 45 sec.

tested output

enter hour of t_1 : 1

enter min of t_1 : 23

enter sec of t_1 : 10

enter hour of t_2 : 3

enter min of t_2 : 22

enter sec of t_2 : 31

4hr 45min 41sec