Pandas :-

* Powerful open source tool commonly used by data scientists.
* Provides data structures designed to make working with data easy and intuitive.
* Contains functions that enable you to store, manipulate, and analyze data.

Series

A one-dimensional labeled array that can hold any data type.

Create a Pandas Series :-

```
# import relavant libraries
import numpy as np
import pandas as pd.
```

You can create a pandas Series from any of the following :-

* An array-like object
* A python dictionary
* A scalar value.

Create a pandas Series Containing 8 random integers from -10 to 10 inclusive.

```
pd.Series(np.random.randint(-10, 11, size=8))
```

Output:-

```
0    10
1    -9
2    -1
3    -2
4     3
5    -7
6     7
7    -5
dtype: int64
```

Create a pandas Series Containing 8 random integers from -10 to 10 inclusive with index being a through h.

```
pd.Series(np.random.random(-10, 11, size=8),
          index=['a','b','c','d','e','f','g','h'])
```

output:-

```
a   -6
b   -8
c   -6
d    7
e    5
f   -10
g   -1
h    6
dtype: int64.
```

Create a pandas series from a python dictionary

```
pd.Series ( {'mon': True, 'Tue': false, 'wed': True,
             'Thu': false, 'Fri': True, 'sat': false, 'sun': True})
```

output:-

```
Mon    True
Tue    false
wed    True
Thu.   false
Fri    True
Sat    false
Sun    True.
dtype: bool
```

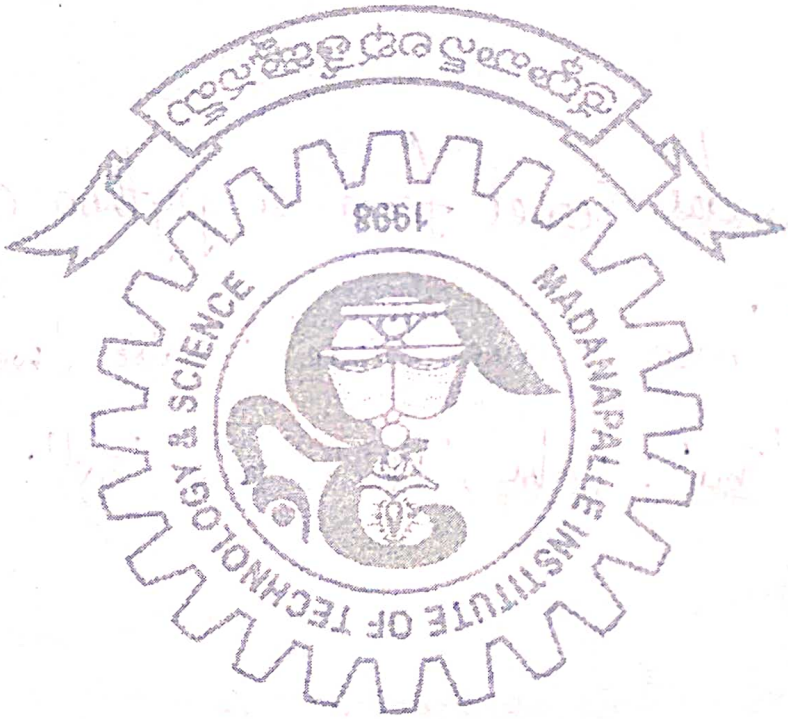
Create a pandas Series from a Scalar value to represent the maximum number of points students can earn in each of 10 exams for a particular course.

```
Pd.Series(150, index=np.arange(1,11))
```

output :-

1	150
2	150
3	150
4	150
5	150
6	150
7	150
8	150
9	150
10	150

dtype: int64



Pandas Library

* powerful open source tool commonly used by data scientists.
* Supports data manipulation and data analysis.

Data frame :-

A 2-dimensional labeled data structure with columns that can hold any data type.

Create a Pandas Data Frame :-

```
#import relevant libraries.
import numpy as np
import pandas as pd.
#read grades.csv into a pandas dataframe & save
 the dataframe in a variable.
grades = pd.read_csv('grades.csv')
#display grades
grades
```

#output:- up to last based on csv file. (large data)

:

display first few rows of grades.

```
grades.head()
```

#output:-

	exam	studentid	grade
0	1	1	86.0
1	1	2	65.0
2	1	3	70.0
3	1	4	98.0
4	1	5	89.0

You call the pandas DataFrame function on any of the following:-

=> A python Dictionary of one-dimensional Numpy Arrays, lists, Dictionaries, or series.

=> A multi-Dimensional Numpy array.

=> A structured or Record Array.

=> A Series

=> Another DataFrame.

#Create a python dictionary of series & save in a variable named points

```
points = {'player1': pd.Series([15,10,20,25],
          index=['game1','game2','game3','game4']),
          'player2': pd.Series([10,15,23,27],
          index=['game1','game2','game3','game4'])}
```

#Create a pandas dataframe from points.

```
pd.DataFrame(points)
```

#output :-

	player1	player2
game1	15	10
game2	10	15
game3	20	23
game4	25	27

MITS

Create a python dictionary of lists & save in a variable named sales.

```
Sales = {'foodTruck1': [216, 275, 203, 210, 315, 402, 380],
         'foodTruck2': [374, 90, 95, 115, 130, 150, 140]}
```

Create a pandas dataframe from sales with index being ~~dat~~ day1 to day7.

```
pd.DataFrame(sales, index=['day1', 'day2', 'day3', 'day4',
                           'day5', 'day6', 'day7'])
```

#Output

	foodTruck1	foodtruck2
day1	216	374
day2	275	90
day3	203	95
day4	210	115
day5	315	130
day6	402	150
day7	380	140.

Create a multi-dimensional numpy array & save in a variable name passangers.

```
passangers = np.array([[20, 40, 60, 80], [15, 30, 45, 60], [10, 20, 30, 40]])
```

Create a pandas Data Frame from passengers. with index being plane1 through plane3 and columns being infants, children, adults, seniors.

```
pd.DataFrame(passangers, index = ['plane1', 'plane2', 'plane3'], columns = ['infants', 'children', 'adults', 'seniors'])
```

output:-

	infants	children	adults	seniors
plane1	20	40	60	80
plane2	15	30	45	60
plane3	10	20	30	40

Select Data Subsets from Pandas Objects

MITS

display first few rows of grades.

grades.head() → top 5 rows will print.

#output:-

	exam	studentid	grade
0	1	1	86.0
1	1	2	65.0
2	1	3	70.0
3	1	4	98.0
4	1	5	89.0

Select exam column and grade column from grades dataframe, result will be a pandas dataframe containing just those columns.

```
grades.loc[:, ['exam', 'grade']]
```

Select exam column from grades dataframe result will be a pandas dataframe containing just that column.

```
grades.loc[:, ['grade']]
```

#Select row 0 from grades dataframe such that result is a pandas Series.

```
grades.iloc[0]
```

#output:-

```
exam          1.0
studentid     1.0
grade        86.0
Name: 0, dtype: float64
```

#Select row 0 from grades dataframe such that result is a pandas dataframe.

```
grades.iloc[[0],:]
```

#output:-

	exam	studentid	grade
0	1	1	86.0

#Select row 0 and row 10 from grades dataframe result will be a pandas data frame.

```
grades.iloc[[0,10],:]
```

#output

	exam	studentid	grade
0	1	1	86.0
10	2	1	79.0

MITS

Select item at row4 column 2 from grades data frame.

```
grades.iloc[4, 2]
```

#output

89.0

Select column 0 and column2 from grades dataframe result will be a pandas data frame.

```
grades.iloc[:, [0, 2]]
```

Select row0 and row2 from grades dataframe.

```
grades.iloc[[0, 2], :]
```

#output:

	exam	student id	grade
0	1	1	86.0
2	1	3	70.0

Seled rows 35 and 45 and columns 0 and 2 from grades dataframe, result will be a pandas data frame.

```
grades.iloc[[35, 45], [0, 2]]
```

#output:-

	exam	grade
35	4	84.0
45	5	94.0

#Select every row from grades dataframe for which entry in grade column is atleast 70.0.

```
grades[grades['grade'] >= 70.0]
```

OR

select all data representing student grades on exam no. 5 that were atleast 70%. in other words, select every row from grades dataframe for which entry in exam column is 5 and entry in grade column is atleast 70.0.

```
grades[(grades['exam']==5) & (grades['grade']>=70.0)]
```

Modify Pandas Objects

MITS

```
#display grades.
grades.

#fill missing values in grade column with zeros
grades['grade'] = grades['grade'].fillna(0)
#display grades
grades
#drop student_id column from grades
grades = grades.drop(columns=['student_id'])
grade

#rename exam column --- change that column's
label from 'exam' to 'exam#'
grades = grades.rename(columns={'exam':'exam#'})
```

Combine Data from Pandas Objects.

```
#display exam1_grades
exam1_grades
#display exam2_grades
exam2_grades
#Merge exam1_grades and exam2_grades on 'SID'
#result will be a new pandas data frame.
pd.merge(exam1_grades, exam2_grades, on='SID')
```

```
#display sid_1_to_7
sid_1_to_7.
#display sid_8_to_10
sid_8_to_10.
#Concatenate sid_1_to_7 and sid_8_to_10 along the rows (along axis=0)
#result will be a new pandas data frame
pd.concat([sid_1_to_7, sid_8_to_10], axis=0)
```

```
#display exam1 and 2
exam1 and 2
#display exam3
exam3
```

#Concatenate exam1 and2 and exam3 the column along (axis;1)
#result will be a new pandas dataframe.

pd.Concat([exam1and2, exam3], axis=1)

Group Data from Pandas Objects :-

#display grades
grades

#drop exam column from grades, group by Student_id, and Compute mean grade for each Student_id result will be a new pandas Data frame.

grades.drop(columns='exam').groupby('student_id').mean()

#drop student_id column from grades, group by exam, and compute mean grade for each exam result will be a new pandas dataframe.

```
grades.drop(columns='student_id').groupby
('exam').mean()
```