Course Code : MCS-201
Course Title : Programming in C and PYTHON
Assignment Number : PGDCA(I)/201/Assignment/2021
Maximum Marks : 100
Weightage : 30%
Last Date of Submission : 31stMay, 2021 (for January session)
31stOctober, 2021(for July session)

Question1: Write an algorithm, draw a flow chart and write its corresponding C program to convert a decimal number to its equivalent Binary number. **(8 Marks)**

Question2: Write an algorithm and its corresponding C program to generate students' Progress-Report for VIII standard (section of 20 students) of a CBSE school for all its 4 terms. Use Structures concept. Assumptions can be made wherever necessary. **(8 Marks)**

Question 3: Write a C program to generate the following pattern: **(8 Marks)**

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

Question 4: Write a C program to perform the following operation on matrices D = A + (B * C), where A, B and C are matrices of (3 X 3) size and D is the resultant matrix. **(8 Marks)**

Question 5: Write a C program to take a list of N numbers, separate even and odd numbers and put them in two appropriate files (evenfile and oddfile). Use File Handling concept. **(8 Marks)**

SECTION-B (PYTHON-Programming)

Question 6: Write a program in Python to check if a given year (entered by user) is a leap year or not, support your programme with suitable comments to improve readability **(8 Marks)**

Question 7: Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error. If the score is between 0.0 and 1.0, print a grade using the following table **(8 Marks)** 4

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

Question 8: Write a programme in Python to create a package named Area and create 3 module in it named – square, circle and rectangle each having a function to calculate area of square, circle and rectangle respectively. Import the module in separate location and use the functions. **(8 Marks)**

Question 9: Write a program in Python to perform following: **(8 Marks)**

⊡ To find cube of numbers in a list using lambda function.

⊡ To display frequency of each word in a file.

⊡ To display first n lines from a file, where n is given by user.

⊡ To display size of a file in bytes

Question 10: What are Co-routines? How Co-routines support cooperative multi-tasking in python? How Co-routines differ from threads? Compare Subroutines and Co-routines.

SECTION A

Ans -1

Algorithm

- Step 1: Divide the number by 2 through % (modulus operator) and store the remainder in array
- Step 2: Divide the number by 2 through / (division operator)
- Step 3: Repeat the step 2 until number is greater than 0

C Pogram

```c
1. #include<stdio.h>
2. #include<stdlib.h>
3. int main(){
4. int a[10],n,i;
5. system ("cls");
6. printf("Enter the number to convert: ");
7. scanf("%d",&n);
8. for(i=0;n>0;i++)
9. {
10.        a[i]=n%2;
11.        n=n/2;
12.        }
13.        printf("\nBinary of Given Number is=");
14.        for(i=i-1;i>=0;i--)
15.        {
16.        printf("%d",a[i]);
17.        }
18.        return 0;
19.        }
```
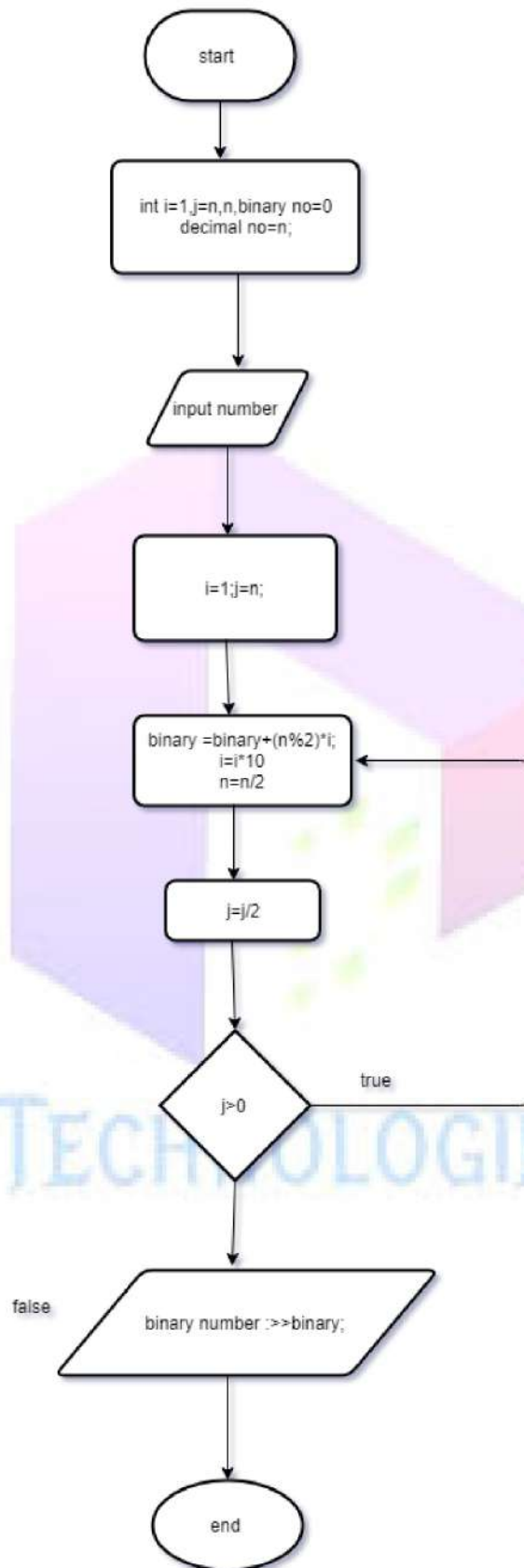
Output:

Enter the number to convert: 5
Binary of Given Number is=101

Flow chart

```
                    ┌──────────┐
                    │   start   │
                    └──────────┘
                          │
                          ▼
              ┌───────────────────────────┐
              │ int i=1,j=n,n,binary no=0 │
              │      decimal no=n;        │
              └───────────────────────────┘
                          │
                          ▼
                  ╱─────────────────╲
                 │   input number    │
                  ╲─────────────────╱
                          │
                          ▼
              ┌───────────────────────────┐
              │         i=1;j=n;          │
              └───────────────────────────┘
                          │
                          ▼
              ┌───────────────────────────┐
              │  binary =binary+(n%2)*i;  │◄──────┐
              │           i=i*10          │       │
              │           n=n/2           │       │
              └───────────────────────────┘       │
                          │                        │
                          ▼                        │
                   ┌────────────┐                  │
                   │    j=j/2   │                  │
                   └────────────┘                  │
                          │                        │
                          ▼                        │
                      ◇─────────◇          true    │
                     ◇   j>0    ◇─────────────────┘
                      ◇─────────◇
                          │
         false            ▼
                 ╱─────────────────────────╲
                │ binary number :>>binary;  │
                 ╲─────────────────────────╱
                          │
                          ▼
                    ┌──────────┐
                    │    end    │
                    └──────────┘
```

Ans 2.

```c
#include<stdio.h>
struct student
{
int roll;
char name[50];
int result[4][9];
};

struct student st[20];
int i, j, pos=-1;



void insertresult();
void display();



void main()
{
int choice;
while(1)
{
printf("\n Press 1 for insert result");
printf("\n Press 2 for display result");
printf("\n Press 3 for exit");
printf("\n Enter your choice... ");
scanf("%d",&choice);
switch(choice)
{
        case 1:
        insertresult();
        break;

        case 2:
        display();
        break;

        case 3:
        exit(0);
        default:
        printf("\n Wrong Input");
          }
      }
}

void insertresult()
{
pos++;
printf("\n Enter the student roll number... ");
scanf("%d",&st[pos].roll);
```
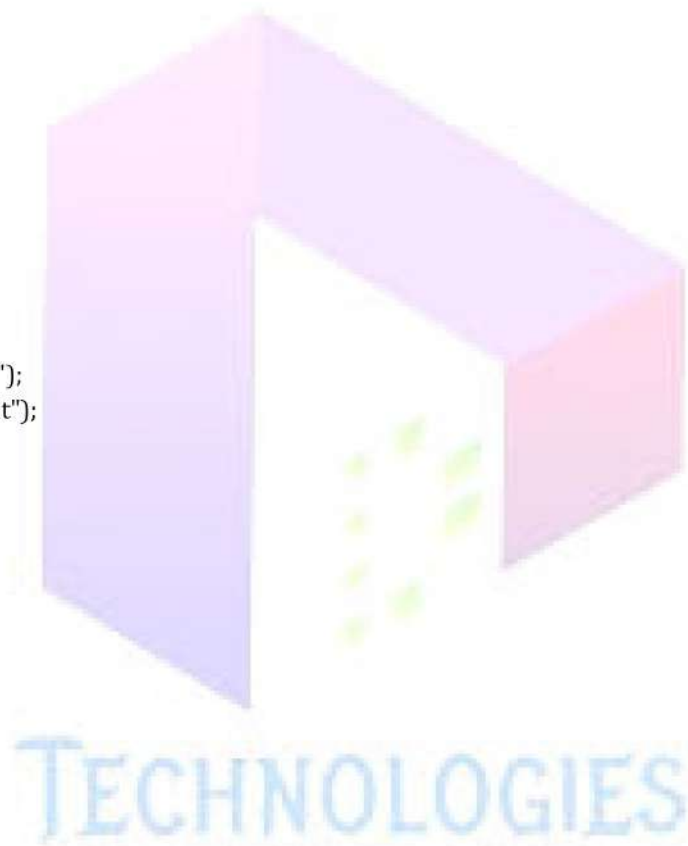
```c
printf("\n Enter the student name... ");
fflush(stdin);
gets(st[pos].name);
for(i=0;i<4;i++)


{
printf("\n Enter the Marks for Term %d",(i+1));
printf("\n English 1 : ");
scanf("%d",&st[pos].result[i][0]);
printf("\n English 2 : ");
scanf("%d",&st[pos].result[i][1]);
printf("\n Mathematics : ");
scanf("%d",&st[pos].result[i][2]);
printf("\n Physics : ");
scanf("%d",&st[pos].result[i][3]);
printf("\n Chemistry : ");
scanf("%d",&st[pos].result[i][4]);
printf("\n Biology : ");
scanf("%d",&st[pos].result[i][5]);
printf("\n Computer : ");
scanf("%d",&st[pos].result[i][6]);
printf("\n History : ");
scanf("%d",&st[pos].result[i][7]);
printf("\n Geography : ");
scanf("%d",&st[pos].result[i][8]);
    }
}


void display()
{
int n, found=0;
printf("\n Enter the roll number... ");
scanf("%d",&n);
for(i=0;i<=pos;i++)
{
if(st[i].roll == n)
{
found=1;
printf("\n\n %s",st[i].name);
printf("\n\n Subject \t Term-1 \t Term-2 \t Term-3 \t Term-4");
printf("\n\n English 1 ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][0]);
printf("\n\n English 2 ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][1]);
printf("\n\n Mathematics ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][2]);
printf("\n\n Physics ");
```
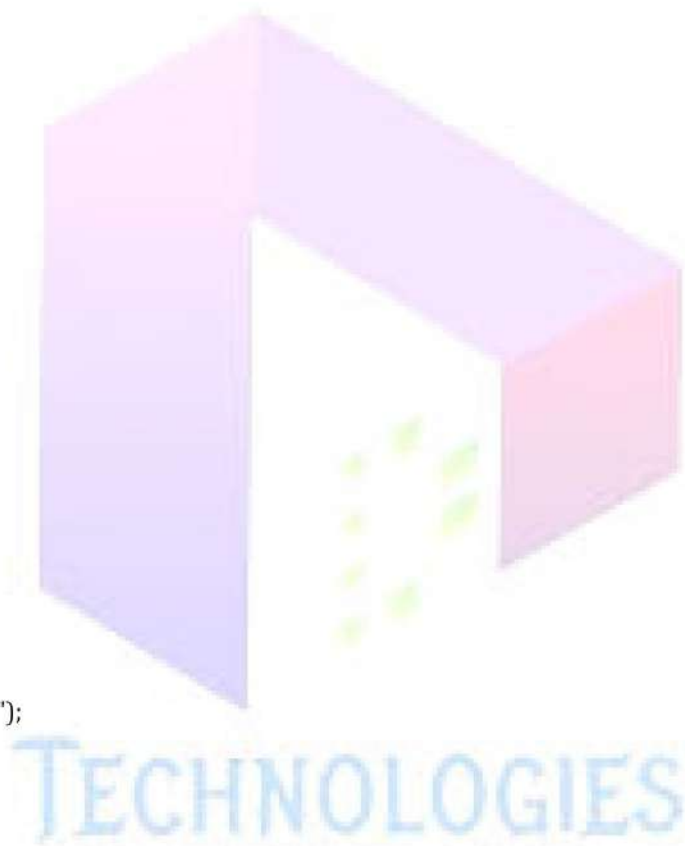
```
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][3]);
printf("\n\n Chemistry ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][4]);
printf("\n\n Biology ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][5]);
printf("\n\n Computer ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][6]);
printf("\n\n History ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][7]);
printf("\n\n Geography ");
for(j=0;j<4;j++)
printf("\t\t %d ",st[i].result[j][8]);
break;
}
}
if(found == 0)
printf("\n Given roll number not exist");
}
```

Ans 3.

Half Pyramid of Numbers

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

C Program

```c
#include <stdio.h>
int main() {
  int i, j, rows;
  printf("Enter the number of rows: ");
  scanf("%d", &rows);
  for (i = 1; i <= rows; ++i) {
    for (j = 1; j <= i; ++j) {
      printf("%d ", j);
    }
    printf("\n");
  }
  return 0;
}
```

Output :

Enter Number Of Rows You Want: 5

```
1
1 2
1 2 3
1 2 3 4
```

1 2 3 4 5

Ans4.

```c
#include<stdio.h>
void main()
{
int i, j, k, a[3][3], b[3][3], c[3][3], d[3][3];
// Input Matrix A
printf("\n Enter the element of matrix A");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("\n Enter the value of A[%d][%d] ",i,j);
scanf("%d",&a[i][j]);
}
}
// Input Matrix B
printf("\n Enter the element of matrix B");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("\n Enter the value of B[%d][%d] ",i,j);
scanf("%d",&b[i][j]);
}
}
// Input Matrix C
printf("\n Enter the element of matrix C");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("\n Enter the value of C[%d][%d] ",i,j);
scanf("%d",&c[i][j]);
// Initialize Matrix D
d[i][j]=0;
}
}
// Matrix Multilication
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
for(k=0;k<3;k++)
{
d[i][j]=(d[i][j] + (b[i][k] * c[k][j]));
}
}
}
// Matrix Addition
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
d[i][j]=d[i][j]+a[i][j];
}
}
```

// Print Matrix D

```
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("\t%d",d[i][j]);
}
printf("\n");
}
}
```

Output

Enter the Element of First Matrix
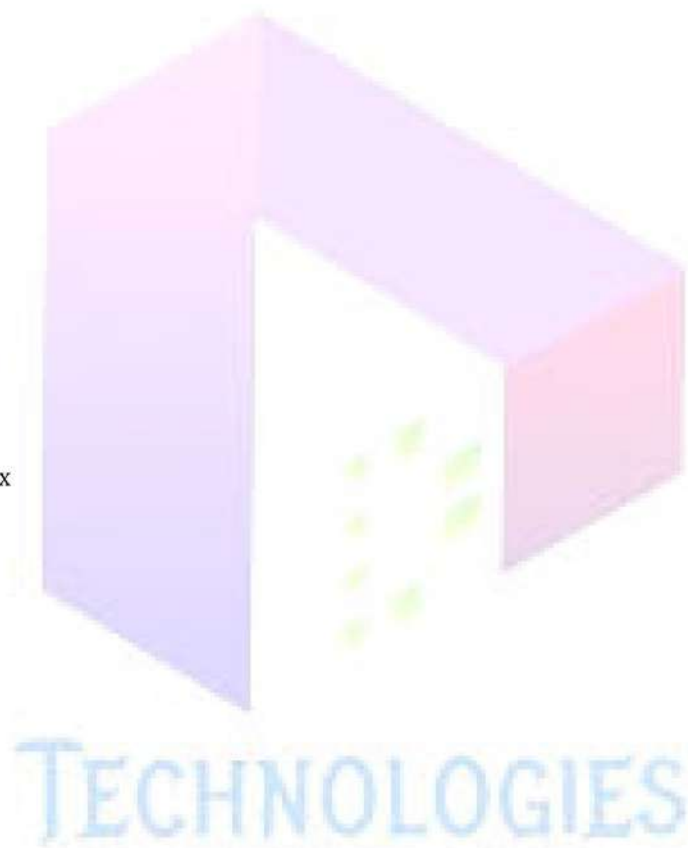Enter the value of A[0][0] : 6

Enter the value of A[0][1] : 9

Enter the value of A[0][2] : 1

Enter the value of A[1][0] : 7

Enter the value of A[1][1] : 2

Enter the value of A[1][2] : 4

Enter the value of A[2][0] : 8

Enter the value of A[2][1] : 3

Enter the value of A[2][2] : 7

Enter the Element of Second Matrix
Enter the value of B[0][0] : 1

Enter the value of B[0][1] : 4

Enter the value of B[0][2] : 2

Enter the value of B[1][0] : 3

Enter the value of B[1][1] : 2

Enter the value of B[1][2] : 5

Enter the value of B[2][0] : 4

Enter the value of B[2][1] : 6

Enter the value of B[2][2] : 3

Enter the Element of Third Matrix
Enter the value of C[0][0] : 2

Enter the value of C[0][1] : 6

Enter the value of C[0][2] : 3

Enter the value of C[1][0] : 8

Enter the value of C[1][1] : 1

Enter the value of C[1][2] : 5

Enter the value of C[2][0] : 6

Enter the value of C[2][1] : 2

Enter the value of C[2][2] : 4

52	23	32
59	32	43
82	39	61

Ans5.

Answer : -

```c
#include<stdio.h>
#include
void main()
{
FILE *even, *odd;
int n, array[100], i;
printf("\n How many numbers you want to take... ");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter the number... ");
scanf("%d",&array[i]);
}
even=fopen("evenfile","w");
odd=fopen("oddfile","w");
for(i=0;i<n;i++)
{
if(array[i]%2 == 0)
putw(array[i],even);
else
putw(array[i],odd);
}
fclose(even);
fclose(odd);
}
```

SECTION B

Ans-6

Program

```python
year = int(input("Enter a year: "))
if (year % 4) == 0:
  if (year % 100) == 0:
    if (year % 400) == 0:
      print("{0} is a leap year".format(year))
    else:
      print("{0} is not a leap year".format(year))
  else:
    print("{0} is a leap year".format(year))
else:
  print("{0} is not a leap year".format(year))
```

output

output-1

Enter a year: 2021

is not a leap year

>

Output-2

Enter a year: 2020

2020 is a leap year

>

Ans-7

```python
score = input("Enter Score: ")
s =  float(score)
x = 'Error'
if s >= 0.9:
        x = 'A'
elif s >=0.8:
        x='B'
elif s >=0.7:
```

```
        x='C'
elif s >= 0.6:
        x='D'
elif s < .6:
        x ='F'
else:
        x ="Out of Range"
print (x)
```

output-1

Enter Score: 0.5

F

>

Output-2

Enter Score: 0.9

A

>

Output-3

Enter Score: 0.7

C

>

Ans8.

```
#area calculator
def areacalculator():
  _input_ = input("Enter the shape you want to calculate area of: ")
  area = 0
  pie = 3.14
  if _input_ == "Square":
     side = int(input("Enter the value of side: "))
     area = area + (side ** 2)
  elif _input_ == "Circle":
     radius = int(input("Enter the value of radius: "))
```

```
    area = area + (2 * pie * radius)
  elif _input_ == "Rectangle":
    length = int(input("Enter the value of length: "))
    width = int(input("Enter the value of length: "))
    area = area + (length * width)
   else:
    print ("Select a valid shape")
  print ("%.2f" % area)


areacalculator()
```

Ans9.

➢ **To find cube of numbers in a list using lambda function.**

Python Code:-

```
# Initializing list
l = [1, 2, 3, 4]


res = list(map(lambda x: x ** 3, l))
print(res)
```

Output:-

[1, 8, 27, 64]

>

➢ **To display frequency of each word in a file.**

Python Code:-

```
# Python code to find frequency of each word
def freq(str):

  # break the string into list of words
  str = str.split()
  str2 = []

  # loop till string values present in list str
  for i in str:

    # checking for the duplicacy
    if i not in str2:

      # insert value in str2
```

```python
    str2.append(i)

  for i in range(0, len(str2)):

    # count the frequency of each word(present
    # in str2) in str and print
    print('Frequency of', str2[i], 'is :', str.count(str2[i]))

def main():
  str ='apple mango apple orange orange apple guava mango mango'
  freq(str)

if __name__=="__main__":
  main()        # call main function
```

Output:-

```
Frequency of apple is : 3
Frequency of mango is : 3
Frequency of orange is : 2
Frequency of guava is : 1
>
```

➢ **To display first n lines from a file, where n is given by user.**

Program Code :-

```python
f = open (" C:/Users/DalalTechnologies/Desktop/Dalaltechnologies.text "," r ")
n = int(input("How many lines you want to print : "))
if n<=0:
 print("please enter correct value ")
else:
 for i in range(1,n+1):
   line =f.readline()
   print(line)
```

Output

How many lines you want to print : 5

Thank you for Contacting Us.

For Solved Assignments Visit - https://DalalTechnologies.in

For Making Graphics Designs Visit- https://dalaltechnologies.com

For Any Other Queries Please Leave Your Message Or Call Us On - 01135556974 , Mob- 9654400726

Or Mail Us - admin@dalaltechnologies.com

> **To display size of a file in bytes**

Python Code:-

```
def p_Size():
    f=open("dalaltechnologies.txt", "r")
    str=f.read()
    size=len(str)
    print("size of dalaltechnologies.txt file =" ,size,"Bytes")
p_Size()
```

Output:-

The size of dalaltechnologies.txt is : 0 Bytes

Ans10.

A function is a sequence of instructions packed as a unit to perform a certain task. When the logic of a complex function is divided into several self-contained steps that are themselves functions, then these functions are called helper functions or **subroutines**.

Subroutines in Python are called by **main function** which is responsible for coordination the use of these subroutines. Subroutines have single entry point.
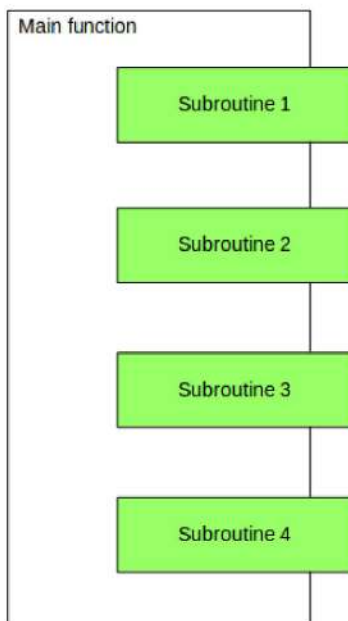

Coroutines are generalization of subroutines. They are used for cooperative multitasking where a process voluntarily **yield** (give away) control periodically or when idle in order to enable **multiple** applications to be run simultaneously. The difference between coroutine and subroutine is :

- Unlike subroutines, coroutines have many entry points for suspending and resuming execution. Coroutine can suspend its execution and transfer control to other coroutine and can resume again execution from the point it left off.
- Unlike subroutines, there is no main function to call coroutines in particular order and coordinate the results. Coroutines are cooperative that means they link together to form a pipeline. One coroutine may consume input data and send it to other which process it. Finally there may be a coroutine to display result.
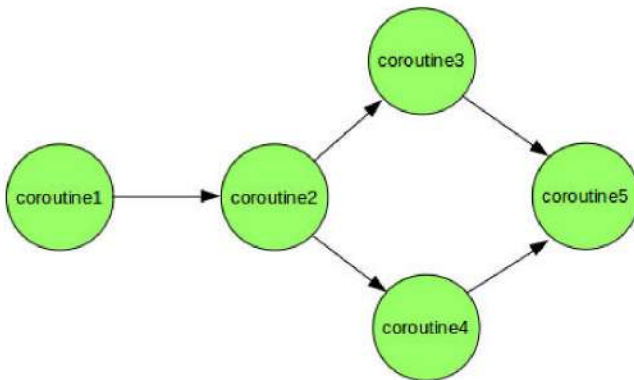
Coroutine Vs Thread

Now you might be thinking how coroutine is different from threads, both seems to do same job.
In case of threads, it's operating system (or run time environment) that switches between threads according to the scheduler. While in case of coroutine, it's the programmer and programming language which decides when to switch coroutines. Coroutines work cooperatively multi task by suspending and resuming at set points by programmer.

Python Coroutine

In Python, coroutines are similar to generators but with few extra methods and slight change in how we use yield statement. Generators produce data for iteration while **coroutines can also consume data.**

In Python 2.5, a slight modification to the yield statement was introduced, now yield can also be used as *expression*. For example on the right side of the assignment –

line = (yield)
whatever value we send to coroutine is captured and returned by **(yield)** expression.
A value can be send to the coroutine by **send()** method.

Execution of coroutine is similar to the generator. When we call coroutine nothing happens, it runs only in response to the **next()** and **send()** method. This can be seen clearly in above example, as only after calling **__next__()** method, out coroutine starts executing. After this call, execution advances to the first yield expression, now execution pauses and wait for value to be sent to *corou* object. When first value is sent to it, it checks for prefix and print name if prefix present. After printing name it goes through loop until it encounters *name = (yield)* expression again.

Closing a Coroutine

Coroutine might run indefinitely, to close coroutine **close()** method is used. When coroutine is closed it generates **GeneratorExit** exception which can be catched in usual way. After closing coroutine, if we try to send values, it will raise **StopIteration** exception.