

## CLASS-03

### Nested Loops & Pattern Printing

---

nested loop basic

```
#include<stdio.h>

//nested loops

int main()
{

// nested for loop
    for(int i=1 ; i <= 5 ; i++) //outer loop for rows
    {
        for(int j=1 ; j<=3 ; j++) //inner loop for columns
        {
            // printf("i = %d, j = %d\n",i,j); //print row and col
        }

        // printf("\n"); //new line after one row
    }

/*

**** // row -> 3 , column -> 4
****
****

*/
int row = 3;
int column = 4;
```

```

for(int i = 1 ; i<=row ; i++) //outer loop -> row
{
    for(int j = 1 ; j<=column ; j++) //inner loop -> column
    {
        printf("*"); //print star
    }

    printf("\n"); //new line after row
}

}

```

### **explanation**

outer loop = rows

inner loop = columns

first part shows how nested loops work

second part prints rectangle pattern (3 rows, 4 columns of stars)

pyramid + inverted pyramid

```

#include<stdio.h>

int main(){
    int row= 4;
    /*

        *
       ***
      *****
     ********

    */

    for(int i =1 ;i<=row ;i++) //outer loop for rows
    {

```

```

        for(int space = 1 ; space <= row - i ; space++) //spaces
        {
            // printf(" ");
        }

        for(int star = 1; star <= 2*i - 1 ; star++) //stars
        {
            //printf("%d",star);
        }

        //printf("\n");
    }

for(int i =row ;i>=1 ;i--) //outer loop reverse for inverted pyramid
{
    for(int space = 1 ; space <= row - i ; space++) //spaces
    {
        printf(" ");
    }

    for(int star = 1; star <= 2*i - 1 ; star++) //stars
    {
        printf("*");
    }

    printf("\n");
}
}

```

### **explanation**

first loop prints pyramid

spaces decrease, stars increase

second loop prints inverted pyramid

spaces increase, stars decrease

half pyramid with star/number/alphabet

```
#include<stdio.h>

int main(){
    /*

*
**
***
****
*****

*/
    int row = 5;
    char alphabet = 'A';
    for(int i=1 ; i<=5 ;i++) //rows
    {
        for(int j = 1 ; j<=i ; j++) //columns
        {
            // printf("*"); //stars
            // printf("%c",alphabet); //alphabets
            // printf("%d",j); //numbers
        }
        // alphabet = alphabet + 1;
        // printf("\n");
    }

    alphabet = 'A';
    for(int i=5 ; i>=1 ;i--) //reverse rows
    {
        for(int j = 0 ; j<i ; j++) //columns
        {
            //printf("*");
```

```

        // printf("%c",alphabet+j);
        // printf("%d",j);
    }
    //alphabet = alphabet + 1;
    printf("\n");
}
}

```

## explanation

first loop makes half pyramid increasing

second loop makes half pyramid decreasing

can print stars, numbers, or alphabets

## diamond pattern

```

#include<stdio.h>

int main(){
int n;
scanf("%d",&n);
/*

    *
   ***
  *****
 *****

*/

for(int i =1 ;i<=n ;i++) //pyramid part
{
    for(int space = 1 ; space <= n - i ; space++)
    {
        printf(" ");
    }

```

```

        for(int star = 1; star <= 2*i - 1 ; star++)
        {
            printf("*");
        }

        printf("\n");
    }

//inverted pyramid
for(int i =n-1 ;i>=1 ;i--)
{
    for(int space = 1 ; space <= n - i ; space++)
    {
        printf(" ");
    }

    for(int star = 1; star <= 2*i - 1 ; star++)
    {
        printf("%d",star);
    }

    printf("\n");
}
}

```

### **explanation**

first loop prints pyramid

second loop prints inverted pyramid

together looks like diamond

top has stars, bottom uses numbers