1.

**public** **class** AmcatPattern {

**public** **static** **void** main(String[] args) {

Scanner scan = **new** Scanner(System.*in*);

System.*out*.println("Enter your number for the pattern\n");

**int** n = scan.nextInt();

**int** number=n;

**for**(**int** i=1;i<=n;i+=2)

{

**int** k=(i-1)\*n+1;

**for**(**int** j=0;j<n-1;j++)

{

System.*out*.print(k+"\*");

k++;

}

System.*out*.print(k);

System.*out*.println("");

}

**if**(n%2!=0)

{

number=n-1;

}

**for**(**int** i=number;i>0;i-=2)

{

**int** k=(i-1)\*n+1;

**for**(**int** j=0;j<n-1;j++)

{

System.*out*.print(k+"\*");

k++;

}

System.*out*.print(k);

System.*out*.println("");

}

scan.close();

}

}

**3.**

**public** **class** AmcatTest {

**public** **static** **void** main(String[] args) {

Scanner scan = **new** Scanner(System.*in*);

System.*out*.println("Enter your number to print pattern\n");

**int** n = scan.nextInt();

**int** printnumber = 1;

**for** (**int** i = 1; i <= n; i++) {

**if** (i % 2 != 0) {

**for** (**int** j = 1; j <= i; j++) {

System.*out*.print(printnumber + "\*");

printnumber++;

}

} **else** {

**int** even=printnumber;

**for** (**int** j = 1; j <= i; j++) {

System.*out*.print((even+(i-1)) + "\*");

printnumber++;

even--;

}

}

System.*out*.println("");

}

scan.close();

}

}

**Full pyramid**

#include<iostream.h>

int main()

{

int space, rows;

cout <<"Enter number of rows: ";

cin >> rows;

for(int i = 1, k = 0; i <= rows; ++i, k = 0)

{

for(space = 1; space <= rows-i; ++space)

{

cout <<" ";

}

while(k != 2\*i-1)

{

cout << "\* ";

++k;

}

cout << endl;

}

getch();

return 0 ;

}

C++ Strings

* A string is a sequence of character
* C++ does not support built-in string type
* Syntax: char string[] = “Hello C++”;
* The one-dimensional array of characters are called strings.
* This is terminated by a null character ***\0***.

Manipulating string

* strcpy(str1, str2): Copies string str2 into string str1.
* strcat(str1, str2): Concatenates string str2 onto the end of string str1.
* strlen(str1): Returns the length of string str1.
* strcmp(str1, str2): Returns 0 if str1 and str2 are the same; less than 0 if str1<str2; greater than 0 if str1>str2.
* strchr(str1, ch): Returns a pointer to the first occurrence of character ch in string str1.
* strstr(str1, str2): Returns a pointer to the first occurrence of string str2 in string str1.

Important functions supported in string

* append(): This function appends a part of a string to another string
* assign():This function assigns a partial string
* at(): This function obtains the character stored at a specified location
* begin(): This function returns a reference to the start of the string
* capacity(): This function gives the total element that can be stored
* compare(): This function compares a string against the invoking string
* empty(): This function returns true if the string is empty
* end(): This function returns a reference to the end of the string
* erase(): This function removes character as specified
* find(): This function searches for the occurrence of a specified substring
* length(): It gives the size of a string or the number of elements of a string
* swap(): This function swaps the given string with the invoking one