1. what is class & object

2. explain class & object for AC & CAR object

3. what is constructor ,what is real use of constructor

4. waht is constructor overloading & constructor chaining

5. what is the use of this & super keyword

6. what is the use of this() & super() statment

7. what is in inheritance & advantages of inheritance

8. expain type of inheritance

9. explain multiple inheritance & why it is not possible, how to achive multiple inheritance

10.what is death of daimond

11.explain overLoading / overRiding with real time example

12.what is abstraction & abstract

13.how many ways we can acheive abstraction

14.Can we have constructor in abstract class?/**/**YES**

15.Can we have main method in abstract class?/***/Yes***

16.can interface inherit object class**?//NO**

17.Can abstract class inherit object class?**//Yes**

18.which is the first statement inside the default constructor?/**/Super**

19.Can interface have a construtor? **?//NO**

20.Can interface have a main method? **?//NO**

21.Can we achieve multiple inheritance by using a abstract class? **?//NO**

22.Can we achieve multiple inheritance by using a interface? ?**//Yes**

23.Can we achieve loose coupling by using abstract class and interface? /**/Yes**

24.Can we have a private abstract method in abstract class**?//NO**

25.Can we have a static variables inside the interface?/**/Yes**

25.how do you achieve multiple inheritance in java by using interface?

26.Explain the interface with real time example?

26.What is exception?

27.Diffrence betweem exception and error?

28.Explain the types of exception?

29.what is the purpose of try and catch block?//To HANDLE EXCEPTION

30.can we have a nested try blocks?**//Yes**

31.can we have a more than one finally block in a single program?

32.what is the diffrence between throw and throws and throwable?

33.what are custom exception?How to write custom excpetion?

35.which is supermost class of exception?

36.what are the inbuilt class present in java?

37.what are the inbuilt methods are present in java?

38.what is the diffrence between final,finally,finalize?

39.List out the exceptions are available in java?

40.Difference between array and collection?

41.difference between array and arraylist?

42.difference between array and linkedlist?

43.difference between arraylist and linkedlist?

44.difference between arraylist and vector?

45.difference between list and set?

46.difference between set and map?

47.difference between hashset and linkedhashset?

48.difference between treeset and treemap?

49.difference between comparable and comparator?

50.how many cursors are available in java?

51.how arraylist is implemented?

52.what is the default size of arraylist?

53.difference between collection and collections?

54.what is the role of iterator in collection framework?

55.difference between hashmap and hashtable?

56. what is early binding and late binding?

56.what is wrapper class, List of the Wrapper class avilable in Java

57.What is boxing & unboxing

58.how to integer to String

59.how to Stringto Intiger

60.what is premetive casting, types of premetive casting

61.what of object casting expalin with example

62.Explain Run time ploymorphism with upcasting example

63.what is acheive down casting in java

64.What is classCASTING exception & when will get ?

65.what is Object class in Java

66.explain inbuilt class avilable in OBJECT class

67.can we create object array? what is the use of Object array

68.explain toString(). eqaul(), hashcode() mtd in Object class

69. What is String class , why is required

70. can we create object to final String class

71. how many ways to create String object

72. why String is immutable , explain with program

73. difference between Constant pool & non constant pool in String class

74. differce between equals() and == and compareTo() method in java?

75. Difference between equals() and contains() method?

76.Diffrence between stringbuffer and stringbuilder?

77.Explain the inbuilt methods of String class?

78.Explain public sttaic void main(String[] args)?

79.Explain system.out.println()?

80.Does arrays are objects in java?

81.What is two dimensional array?

82.What is jagged array?

83.Can we declare array as a final?

84.What is marker interafce?

85.what is serialization?

86.what is deserialisation?

87.which class do we need to use for a serialisation?

88.what are nested class?

89.Can we change the sequence of public static in main method?

90.what are commandline arguments?

91.what are threads?how many ways to create a thread?

92.what is synchronisation?

93.what is singleton class?

94.why singleton class is required?explain with real time example?

95.How to declare ,create, and initialise array in single line?

96.what are the disadvantages of array?

97.which is the default package in java?

98.what is the use of package?

99.what is the use of import keyword?

100.what are inbuilt packages are available in java?

101.Explain oops concepts with real time examples?

102. diff b/n static and non-static.

Programs

1.W.A.P PALINDROME?

2.w.a.p for reverse number?

3.armstrong number?

4.prime number or not?

5.w.a.p to generate prime numbers between 1 to 1000?

6.w.a.p strong number?

7.perfect number?

8.summation of odd and even numbers?

9.fibonacci?

10.factorial?

11.sum of digits?

12.Sum of 'n' natural numbers?

13.bubblesort?

14.w.ap. to display 1 to 100 without using any loop?

15.Can a method return more than one value?if not how to return?

16.sum of array elements?

17.binary search?

18.addition of two matrix?

19.transpose matrix?

20.contigious array?

21.w.a.p to find biggest and least elemet is present in the array?

22.w.a.p to reverse an integer array and display?

23.w.a.p to find duplicate elemenst in array?

25.reverese string?

26.reverse sentence?

27.palindrome in string?

28.w.a.p to find length of the string without using length fuction?

29.w.a.p to count number of vowels,space and digits present in the array?

30.w.a.p to find repeated characters are present instring?

31.w.a.p to find non-repeated characters in String?

32.anagram?

33.w.a.p tp remove the space which is present in the sentence?

34.w.a.p to find to count the number of words present in the text file?

35.w.a.p to count commented line number available in textfile?

36.All pattern programs?

37.w.a.p to find out whether the year is leap or not?

Question-1: Write code to filter duplicate elements from an array and print as a list?

```
package simple.test;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

public class findDuplicates {

        public static void main(String[] args) {

                ArrayList<String> list = new ArrayList<String>();

                // Form a list of numbers from 0-9.
                for (int i = 0; i < 10; i++) {
                        list.add(String.valueOf(i));
                }

                // Insert a new set of numbers from 0-5.
```

```
		for (int i = 0; i < 5; i++) {
			list.add(String.valueOf(i));
		}

		System.out.println("Input list : " + list);
		System.out.println("\nFiltered duplicates : " +
processList(list));
	}

	public static Set<String> processList(List<String>
listContainingDuplicates) {

		final Set<String> resultSet = new HashSet<String>();
		final Set<String> tempSet = new HashSet<String>();

		for (String yourInt : listContainingDuplicates) {
			if (!tempSet.add(yourInt)) {
				resultSet.add(yourInt);
			}
		}
		return resultSet;
	}
}
```

Question-2: Write code to sort the list of strings using Java collection?

```
package simple.test;

import java.util.Arrays;

public class sortStrings {

	public static void main(String[] args) throws Exception {

		String[] inputList = { "Jan", "Feb", "Mar", "Apr", "May",
"Jun", "Jul",
				"aug", "Sep", "Oct", "nov", "Dec" };

		// Display input un-sorted list.
		System.out.println("-------Input List-------");
		showList(inputList);

		// Call to sort the input list.
		Arrays.sort(inputList);

		// Display the sorted list.
		System.out.println("\n-------Sorted List-------");
		showList(inputList);

		// Call to sort the input list in case-sensitive order.
		System.out.println("\n-------Sorted list (Case-Sensitive)-----
--");
		Arrays.sort(inputList, String.CASE_INSENSITIVE_ORDER);
```

```
                // Display the sorted list.
                showList(inputList);
        }

        public static void showList(String[] array) {
                for (String str : array) {
                        System.out.print(str + " ");
                }
                System.out.println();
        }

}
```

Question-3: Write a function to reverse a number in Java?

```
package simple.test;

public class invertNumber {

        public long doInvert(long number) {

                long invert = 0;
                while (number != 0) {
                        invert = (invert * 10) + (number % 10);
                        number = number / 10;
                }
                return invert;
        }

        public static void main(String args[]) {
                long lnum = 654321;
                invertNumber input = new invertNumber();

                System.out.println("Input value : " + lnum);
                System.out.println("Inverted value : " +
input.doInvert(lnum));
        }
}
```

Further Reading:

- [Top 20 Selenium Coding Tips for Software Testers.]

Question-4: Write a method to check prime no. in Java?

```
package simple.test;

import java.util.Scanner;

public class findPrime {
```

```
	public static void main(String[] args) {
		Scanner scan = new Scanner(System.in);
		System.out.print("Enter an int value : ");
		int input = scan.nextInt();
		if (checkPrime(input)) {
			System.out.println("Input value " + input + " is a
prime number.");
		} else {
			System.out.println("Input value " + input
					+ " is not a prime number.");
		}
	}

	public static boolean checkPrime(int n) {
		if (n <= 1) {
			return false;
		}
		for (int i = 2; i < Math.sqrt(n); i++) {
			if (n % i == 0) {
				return false;
			}
		}
		return true;
	}
}
```

Question-5: Write a Java program to find out the first two max values from an array?

```
package simple.test;

public class findTwoMaxValue {
	public void GetTwoMaxValues(int[] nums) {

		int maxOne = 0;
		int maxTwo = 0;
		for (int n : nums) {
			if (maxOne < n) {
				maxTwo = maxOne;
				maxOne = n;
			} else if (maxTwo < n) {
				maxTwo = n;
			}

		}

		System.out.println("Max1 - " + maxOne);
		System.out.println("Max2 - " + maxTwo);
	}

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

```
                int list[] = { 15, 24, 48, 21, 43, 11, 79, 93 };

                findTwoMaxValue max = new findTwoMaxValue();
                max.GetTwoMaxValues(list);
        }
}
```

Question-6: Write a Java program to find the longest substring from a given string which doesn't contain any duplicate characters?

```
package simple.test;

import java.util.HashSet;
import java.util.Set;

public class findSubstr {

        private Set<String> stringSet = new HashSet<String>();
        private int lstringSet = 0;

        public Set<String> findStr(String input) {

                // Reset instance data.
                stringSet.clear();
                lstringSet = 0;

                // Set a boolean flag on each char's ASCII value.
                boolean[] flag = new boolean[256];
                int j = 0;
                char[] inputCharArr = input.toCharArray();
                for (int i = 0; i < inputCharArr.length; i++) {
                        char c = inputCharArr[i];
                        if (flag[c]) {
                                extractSubString(inputCharArr, j, i);
                                for (int k = j; k < i; k++) {
                                        if (inputCharArr[k] == c) {
                                                j = k + 1;
                                                break;
                                        }
                                        flag[inputCharArr[k]] = false;
                                }
                        } else {
                                flag[c] = true;
                        }
                }
                extractSubString(inputCharArr, j, inputCharArr.length);
                return stringSet;
        }

        private String extractSubString(char[] inputArr, int start, int end) {

                StringBuilder sb = new StringBuilder();
                for (int i = start; i < end; i++) {
```

```
                        sb.append(inputArr[i]);
                }
                String subStr = sb.toString();
                if (subStr.length() > lstringSet) {
                        lstringSet = subStr.length();
                        stringSet.clear();
                        stringSet.add(subStr);
                } else if (subStr.length() == lstringSet) {
                        stringSet.add(subStr);
                }

                return sb.toString();
        }

        public static void main(String a[]) {

                findSubstr substr = new findSubstr();

                System.out
                                .println("Actual Strings ------------ | ----
Longest Non-Repeated Strings");
                System.out.println("Software_Programmer"
                                + "         |         " +
substr.findStr("Software_Programmer"));
                System.out.println("Software_Developer_In_Test"
                                +  "  |         " +
substr.findStr("Software_Developer_In_Test"));
                System.out.println("developers_write_unit_tests"
                                +  " |         " +
substr.findStr("developers_write_unit_tests"));
                System.out.println("javajavbasp.net"
                                +  "             |         " +
substr.findStr("javajavbasp.net"));
        }
}
```

Question-7: Write Java code to get rid of multiple spaces from a string?

```
package simple.test;

import java.util.StringTokenizer;

public class removeExtraSpaces {

    public static void main(String args[]){

        String input = "Try    to    remove   extra   spaces.";
        StringTokenizer substr = new StringTokenizer(input, " ");
        StringBuffer sb = new StringBuffer();

        while(substr.hasMoreElements()){
            sb.append(substr.nextElement()).append(" ");
        }
```

```
        System.out.println("Actual string: " + input);
        System.out.println("Processed string: " + sb.toString().trim());
    }
}
```

Question-8: Write Java code to identify a number as Palindrome?

```
package simple.test;

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class identifyPalindrome {

        public static void main(String[] args) {

                try {
                        BufferedReader object = new BufferedReader(new
InputStreamReader(
                                        System.in));
                        System.out.println("Input number");
                        int inputValue = Integer.parseInt(object.readLine());
                        int n = inputValue;
                        int rev = 0;
                        System.out.println("Input value is : ");
                        System.out.println(" " + inputValue);
                        for (int i = 0; i <= inputValue; i++) {
                                int r = inputValue % 10;
                                inputValue = inputValue / 10;
                                rev = rev * 10 + r;
                                i = 0;
                        }
                        System.out.println("Post reversal : " + " ");
                        System.out.println(" " + rev);
                        if (n == rev) {
                                System.out.print("Input value is a
palindrome.");
                        } else {
                                System.out.println("Input value is not a
palindrome.");
                        }
                } catch (Exception e) {
                        System.out.println("Out of Range.");
                }
        }
}
```

Question-9: Write Java code to swap two numbers without using a temporary variable?

```
package simple.test;

public class smartSwapping {

	public static void main(String args[]) {

		int numX = 10;
		int numY = 20;
		System.out.println("Pre-swapping state:");
		System.out.println("numX value: " + numX);
		System.out.println("numY value: " + numY);

		System.out.println("");

		numX = numX + numY;
		numY = numX - numY;
		numX = numX - numY;
		System.out.println("Post-swapping state:");
		System.out.println("numX value: " + numX);
		System.out.println("numY value: " + numY);
	}
}
```

Question-10: Write a Java program to demonstrate string reverse with and without StringBuffer class?

```
package simple.test;

public class invertString {

	public String invertWithStringBuffer(String str) {

		StringBuffer buffer = new StringBuffer(str);
		buffer.reverse();
		return buffer.toString();
	}

	public String invertWithoutStringBuffer(String str) {

		int length = str.length();
		String original = str;
		String invert = "";
		for (int i = length - 1; i >= 0; i--) {
			invert = invert + original.charAt(i);
		}
		return invert;
	}

	public static void main(String[] args) {

		invertString invertStr = new invertString();

		System.out.println("Inverted String with StringBuffer class: "
```

```
                                    +
invertStr.invertWithStringBuffer("987654321"));

                System.out.println("");

                System.out.println("Inverted String without StringBuffer
class: "
                                    +
invertStr.invertWithoutStringBuffer("kjihgfedcba"));
        }
}
```

Core Java Interview Questions and Answers

1. What is a JVM?

JVM is Java Virtual Machine which is a run time environment for the compiled java class files.

2. Does Java support multiple inheritance?

Java doesn't support multiple inheritance.

3. What is the most important feature of Java?

Java is a platform independent language.

4. What is difference between Path and Classpath?

Path and Classpath are operating system level environment variales. Path is used define where the system can find the executables(.exe) files and classpath is used to specify the location .class files.

5. What are instance variables?

Instance variables are those which are defined at the class level. Instance variables need not be initialized before using them as they are automatically initialized to their default values.

6. What is a pointer and does Java support pointers?

Pointer is a reference handle to a memory location. Improper handling of pointers leads to memory leaks and reliability issues hence Java doesn't support the usage of pointers.

7. What is the return type of the main() method?

Main() method doesn't return anything hence declared void.

8. What are local variables?

Local variables are those which are declared within a block of code like methods. Local variables should be initialized before accessing them.

9. Is Java a pure object oriented language?

Java uses primitive data types and hence is not a pure object oriented language.

10. Is JVM platform independent?

JVM's are not platform independent. JVM's are platform specific run time implementation provided by the vendor.
Java Standard Edition

11. Can a main() method be overloaded?

Yes. You can have any number of main() methods with different method signature and implementation in the class.

12. What is the base class of all classes?

java.lang.Object

13. What do you mean by platform independence?

Platform independence means that we can write and compile the java code in one platform (eg Windows) and can execute the class in any other supported platform eg (Linux,Solaris,etc).

14. Are arrays primitive data types?

In Java, Arrays are objects.

15. What is the difference between a JDK and a JVM?

JDK is Java Development Kit which is for development purpose and it includes execution environment also. But JVM is purely a run time environment and hence you will not be able to compile your source files using a JVM.

16. Does the order of public and static declaration matter in main() method?

No. It doesn't matter but void should always come before main().

17. What is the impact of declaring a method as final?

A method declared as final can't be overridden. A sub-class can't have the same method signature with a different implementation.

18. Can a class be declared as protected?

A class can't be declared as protected. only methods can be declared as protected.

19. How to define a constant variable in Java?

The variable should be declared as static and final. So only one copy of the variable exists for all instances of the class and the value can't be changed also.
static final int PI = 2.14; is an example for constant.

20. Which package is imported by default?

java.lang package is imported by default even without a package declaration.

21. What is the argument of main() method?

main() method accepts an array of String object as argument.

22. Can a source file contain more than one class declaration?

Yes a single source file can contain any number of Class declarations but only one of the class can be declared as public.

23. What is the access scope of a protected method?

A protected method can be accessed by the classes within the same package or by the sub classes of the class in any package.

24. Can a class declared as private be accessed outside it's package?

Not possible.

25. Why is the main() method declared static?

main() method is called by the JVM even before the instantiation of the class hence it is declared as static.

26. What is the purpose of declaring a variable as final?

A final variable's value can't be changed. final variables should be initialized before using them.

27. What is a package?

Package is a collection of related classes and interfaces. package declaration should be first statement in a java class.

28. Can a main() method be declared final?

Yes. Any inheriting class will not be able to have it's own default main() method.

29. I don't want my class to be inherited by any other class. What should i do?

You should declared your class as final. But you can't define your class as final, if it is an abstract class. A class declared as final can't be extended by any other class.

30. Should a main() method be compulsorily declared in all java classes?

No not required. main() method should be defined only if the source class is a java application.

31. When will you define a method as static?

When a method needs to be accessed even before the creation of the object of the class then we should declare the method as static.

32. What is the importance of static variable?

static variables are class level variables where all objects of the class refer to the same variable. If one object changes the value then the change gets reflected in all the objects.

33. How is final different from finally and finalize()?

final is a modifier which can be applied to a class or a method or a variable. final class can't be inherited,final method can't be overridden and final variable can't be changed.

finally is an exception handling code section which gets executed whether an exception is raised or not by the try block code segment.
finalize() is a method of Object class which will be executed by the JVM just before garbage collecting object to give a final chance for resource releasing activity.

34. I want to print "Hello" even before main() is executed. How will you achieve that?

Print the statement inside a static block of code. Static blocks get executed when the class gets loaded into the memory and even before the creation of an object. Hence it will be executed before the main() method. And it will be executed only once.

35. Can we declare a static variable inside a method?

Static variables are class level variables and they can't be declared inside a method. If declared, the class will not compile.

36. Can a abstract class be defined without any abstract methods?

Yes it's possible. This is basically to avoid instance creation of the class.

37. Can you give few examples of final classes defined in Java API?

java.lang.String, java.lang.Math are final classes.

38. Can a abstract class be declared final?

Not possible. An abstract class without being inherited is of no use and hence will result in compile time error.

39. Can you create an object of an abstract class?

Not possible. Abstract classes can't be instantiated.

40. What are the restriction imposed on a static method or a static block of code?

A static method should not refer to instance variables without creating an instance and cannot use "this" operator to refer the instance.

41. Can a Class extend more than one Class?

Not possible. A Class can extend only one class but can implement any number of Interfaces.

42. Class C implements Interface I containing method m1 and m2 declarations. Class C has provided implementation for method m2. Can i create an object of Class C?

No not possible. Class C should provide implementation for all the methods in the Interface I. Since Class C didn't provide implementation for m1 method, it has to be declared as abstract. Abstract classes can't be instantiated.

43. Can a class be defined inside an Interface?

Yes it's possible.

44. Can an Interface extend another Interface?

Yes an Interface can inherit another Interface, for that matter an Interface can extend more than one Interface.

45. What is use of a abstract variable?

Variables can't be declared as abstract. only classes and methods can be declared as abstract.

46. Can an Interface be defined inside a class?

Yes it's possible.

47. What is an Abstract Class and what is it's purpose?

A Class which doesn't provide complete implementation is defined as an abstract class. Abstract classes enforce abstraction.

48. Can an Interface be final?

Not possible. Doing so so will result in compilation error.

49. Can an Interface implement another Interface?

Interfaces doesn't provide implementation hence a interface cannot implement another interface.

50. Why does Java not support operator overloading?

Operator overloading makes the code very difficult to read and maintain. To maintain code simplicity, Java doesn't support operator overloading.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
```

```
1111111
1111122
1111333
1114444
1155555
1666666
7777777
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
```

```
7 6 5 4 3 2 1
7 6 5 4 3 2
7 6 5 4 3
7 6 5 4
7 6 5
7 6
7
```

```
7
7 6
7 6 5
7 6 5 4
7 6 5 4 3
7 6 5 4 3 2
7 6 5 4 3 2 1
```

```
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
```

```
1234567
 234567
  34567
   4567
    567
     67
      7
     67
    567
   4567
  34567
 234567
1234567
```

```
1 2 3 4 5 6 7
 2 3 4 5 6 7
  3 4 5 6 7
   4 5 6 7
    5 6 7
     6 7
      7
     6 7
    5 6 7
   4 5 6 7
  3 4 5 6 7
 2 3 4 5 6 7
1 2 3 4 5 6 7
```

```
1
10
101
1010
10101
101010
1010101
```

```
1010101
0101010
1010101
0101010
1010101
0101010
1010101
```

```
1
2 6
3 7 10
4 8 11 13
5 9 12 14 15
```

```
0000000
0100000
0020000
0003000
0000400
0000050
0000006
```

Pattern 1 :

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
```

Java Program :

```
?

  import java.util.Scanner;
1

2
  public class MainClass
3
  {
4
      public static void main(String[] args)
5     {
6         Scanner sc = new Scanner(System.in);
7
8         //Taking rows value from the user
9
10        System.out.println("How many rows you want in this pattern?");
11
12        int rows = sc.nextInt();
13
14        System.out.println("Here is your pattern....!!!");
15
16        for (int i = 1; i <= rows; i++)
17        {
              for (int j = 1; j <= i; j++)
```

```
18              {
19                  System.out.print(j+" ");
20              }
21
22              System.out.println();
23          }
24
25          //Close the resources
26
27          sc.close();
28      }
29  }
30
31
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7

Pattern 2 :

1
2 2
3 3 3
4 4 4 4

5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7

Java Program :

?

```
1 import java.util.Scanner;
2
3 public class MainClass
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8
9         //Taking rows value from the user
10
11        System.out.println("How many rows you want in this pattern?");
12
13        int rows = sc.nextInt();
14
15        System.out.println("Here is your pattern....!!!");
16
17        for (int i = 1; i <= rows; i++)
18        {
19            for (int j = 1; j <= i; j++)
20            {
21                System.out.print(i+" ");
                }
```

```
22
23              System.out.println();
24          }
25
26          //Close the resources
27
28          sc.close();
29      }
30  }
31
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7

Pattern 3 :

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3

1 2
1

Java Program :

?

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking rows value from the user

        System.out.println("How many rows you want in this pattern?");

        int rows = sc.nextInt();

        System.out.println("Here is your pattern....!!!");

        //Printing upper half of the pattern

        for (int i = 1; i <= rows; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                System.out.print(j+" ");
```

```
            }

            System.out.println();
        }

        //Printing lower half of the pattern

        for (int i = rows-1; i >= 1; i--)
        {
            for (int j = 1; j <= i; j++)
            {
                System.out.print(j+" ");
            }

            System.out.println();
        }

        //Closing the resources

        sc.close();
    }
}
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

Pattern 4 :

1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
   {
4      public static void main(String[] args)
5      {
6          Scanner sc = new Scanner(System.in);
7
8          //Taking rows value from the user
```

```
9
10          System.out.println("How many rows you want in this pattern?");
11
12          int rows = sc.nextInt();
13
14          System.out.println("Here is your pattern....!!!");
15
16          for (int i = rows; i >= 1; i--)
17          {
18              for (int j = 1; j <= i; j++)
19              {
20                  System.out.print(j+" ");
21              }
22
23              System.out.println();
24          }
25
26          //Closing the resources
27
28          sc.close();
29      }
30  }
31
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

Pattern 5 :

7 6 5 4 3 2 1
7 6 5 4 3 2
7 6 5 4 3
7 6 5 4
7 6 5
7 6
7

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
8
9          //Taking rows value from the user
10
11         System.out.println("How many rows you want in this pattern?");
```

```
12          int rows = sc.nextInt();
13
14          System.out.println("Here is your pattern....!!!");
15
16          for (int i = 1; i <= rows; i++)
17          {
18              for (int j = rows; j >= i; j--)
19              {
20                  System.out.print(j+" ");
21              }
22
23              System.out.println();
24          }
25
26          //Closing the resources
27
28          sc.close();
29      }
30  }
31
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
7 6 5 4 3 2 1
7 6 5 4 3 2
7 6 5 4 3
7 6 5 4

7 6 5
7 6
7

Pattern 6 :

7
7 6
7 6 5
7 6 5 4
7 6 5 4 3
7 6 5 4 3 2
7 6 5 4 3 2 1

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
8
9          //Taking rows value from the user
10
11         System.out.println("How many rows you want in this pattern?");
12
13         int rows = sc.nextInt();
14
15         System.out.println("Here is your pattern....!!!");
```

```
16          for (int i = rows; i >= 1; i--)
17          {
18              for (int j = rows; j >= i; j--)
19              {
20                  System.out.print(j+" ");
21              }
22
23              System.out.println();
24          }
25
26          //Closing the resources
27
28          sc.close();
29      }
30  }
31
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
7
7 6
7 6 5
7 6 5 4
7 6 5 4 3
7 6 5 4 3 2
7 6 5 4 3 2 1

Pattern 7 :

7 6 5 4 3 2 1
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
8
9          //Taking rows value from the user
10
11         System.out.println("How many rows you want in this pattern?");
12
13         int rows = sc.nextInt();
14
15         System.out.println("Here is your pattern....!!!");
16
17         for (int i = rows; i >= 1; i--)
18         {
19             for (int j = i; j >= 1; j--)
             {
```

```
20                  System.out.print(j+" ");
21              }
22
23              System.out.println();
24          }
25
26          //Closing the resources
27
28          sc.close();
29      }
30  }
31
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
7 6 5 4 3 2 1
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1

Pattern 8 :

1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
1 2

1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
8
9          //Taking rows value from the user
10
11         System.out.println("How many rows you want in this pattern?");
12
13         int rows = sc.nextInt();
14
15         System.out.println("Here is your pattern....!!!");
16
17         //Printing upper half of the pattern
18
19         for (int i = rows; i >= 1; i--)
20         {
               for (int j = 1; j <= i; j++)
```

```
21              {
22                  System.out.print(j+" ");
23              }
24
25              System.out.println();
26          }
27
28          //Printing lower half of the pattern
29
30          for (int i = 2; i <= rows; i++)
31          {
32              for (int j = 1; j <= i; j++)
33              {
34                  System.out.print(j+" ");
35              }
36
37              System.out.println();
38          }
39
40          //Closing the resources
41
42          sc.close();
43      }
44  }
45
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7

Pattern 9 :

1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
```

```
8         //Taking rows value from the user
9
10        System.out.println("How many rows you want in this pattern?");
11
12        int rows = sc.nextInt();
13
14        System.out.println("Here is your pattern....!!!");
15
16        for (int i = 1; i <= rows; i++)
17        {
18            //Printing first half of the row
19
20            for (int j = 1; j <= i; j++)
21            {
22                System.out.print(j+" ");
23            }
24
25            //Printing second half of the row
26
27            for (int j = i-1; j >= 1; j--)
28            {
29                System.out.print(j+" ");
30            }
31
32            System.out.println();
        }
```

```
33          //Closing the resources
34
35          sc.close();
36      }
37  }
38
39
40
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1

Pattern 10 :

1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
7 6 5 4 3 2 1

Java Program :

?

```
1 import java.util.Scanner;
2
```

```
3 public class MainClass
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8
9         //Taking rows value from the user
10
11        System.out.println("How many rows you want in this pattern?");
12
13        int rows = sc.nextInt();
14
15        System.out.println("Here is your pattern....!!!");
16
17        for (int i = 1; i <= rows; i++)
18        {
19            for (int j = i; j >= 1; j--)
20            {
21                System.out.print(j+" ");
22            }
23
24            System.out.println();
25        }
26
27        //Close the resources

          sc.close();
```

```
28      }
29}
30
31
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
7 6 5 4 3 2 1

Pattern 11 :

?

```
1 1234567
2  234567
3   34567
4    4567
5     567
6      67
7       7
8      67
9     567
10   34567
11  234567
```

1234567

Java Program :

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking rows value from the user

        System.out.println("How many rows you want in this pattern?");

        int rows = sc.nextInt();

        System.out.println("Here is your pattern....!!!");

        //Printing upper half of the pattern

        for (int i = 1; i <= rows; i++)
        {
            //Printing i spaces at the beginning of each row

```

```
21              for (int j = 1; j < i; j++)
22              {
23                  System.out.print(" ");
24              }
25
26              //Printing i to rows value at the end of each row
27
28              for (int j = i; j <= rows; j++)
29              {
30                  System.out.print(j);
31              }
32
33              System.out.println();
34          }
35
36          //Printing lower half of the pattern
37
38          for (int i = rows-1; i >= 1; i--)
39          {
40              //Printing i spaces at the beginning of each row
41
42              for (int j = 1; j < i; j++)
43              {
44                  System.out.print(" ");
45              }

                //Printing i to rows value at the end of each row
```

```
46
47              for (int j = i; j <= rows; j++)
48              {
49                  System.out.print(j);
50              }
51
52              System.out.println();
53          }
54
55          //Closing the resources
56
57          sc.close();
58      }
59  }
60
61
62
63
```

Output :

[?](#)

```
How many rows you want in this pattern?
1
7
2
Here is your pattern....!!!
3
1234567
4
 234567
```

```
5    34567
6     4567
7      567
8       67
9        7
10      67
11     567
12    4567
13   34567
14  234567
15 1234567
16
```

Pattern 12 :

[?]

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

```
 2 3 4 5 6 7
1 2 3 4 5 6 7
```

Java Program :

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking rows value from the user

        System.out.println("How many rows you want in this pattern?");

        int rows = sc.nextInt();

        System.out.println("Here is your pattern....!!!");

        //Printing upper half of the pattern

        for (int i = 1; i <= rows; i++)
        {
            //Printing i spaces at the beginning of each row
```

```
20
21              for (int j = 1; j < i; j++)
22              {
23                  System.out.print(" ");
24              }
25
26              //Printing i to rows value at the end of each row
27
28              for (int j = i; j <= rows; j++)
29              {
30                  System.out.print(j+" ");
31              }
32
33              System.out.println();
34          }
35
36          //Printing lower half of the pattern
37
38          for (int i = rows-1; i >= 1; i--)
39          {
40              //Printing i spaces at the beginning of each row
41
42              for (int j = 1; j < i; j++)
43              {
44                  System.out.print(" ");
```

```
            //Printing i to rows value at the end of each row

            for (int j = i; j <= rows; j++)
            {
                System.out.print(j+" ");
            }

            System.out.println();
        }

        //Closing the resources

        sc.close();
    }
}
```

Output :

```
?
How many rows you want in this pattern?
7
Here is your pattern....!!!
1 2 3 4 5 6 7
```

```
 2 3 4 5 6 7
  3 4 5 6 7
   4 5 6 7
    5 6 7
     6 7
      7
     6 7
    5 6 7
   4 5 6 7
  3 4 5 6 7
 2 3 4 5 6 7
1 2 3 4 5 6 7
```

Pattern 13 :

1
10
101
1010
10101
101010
1010101

Java Program :

?

```
import java.util.Scanner;

public class MainClass
{
```

```
5   public static void main(String[] args)
6   {
7       Scanner sc = new Scanner(System.in);
8
9       System.out.println("How many rows you want in this pattern?");
10
11      int rows = sc.nextInt();
12
13      System.out.println("Here is your pattern....!!!");
14
15      for (int i = 1; i <= rows; i++)
16      {
17          for (int j = 1; j <= i; j++)
18          {
19              if(j%2 == 0)
20              {
21                  System.out.print(0);
22              }
23              else
24              {
25                  System.out.print(1);
26              }
27          }
28
29          System.out.println();
      }
```

```
30          sc.close();
31      }
32  }
33
34
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1
10
101
1010
10101
101010
1010101

Pattern 14 :

1010101
0101010
1010101
0101010
1010101
0101010
1010101

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
```

```
6   {
7       Scanner sc = new Scanner(System.in);
8
9       System.out.println("How many rows you want in this pattern?");
10
11      int rows = sc.nextInt();
12
13      System.out.println("Here is your pattern....!!!");
14
15      for (int i = 1; i <= rows; i++)
16      {
17          int num;
18
19          if(i%2 == 0)
20          {
21              num = 0;
22
23              for (int j = 1; j <= rows; j++)
24              {
25                  System.out.print(num);
26
27                  num = (num == 0)? 1 : 0;
28              }
29          }
30          else
            {
                num = 1;
```

```
31
32                  for (int j = 1; j <= rows; j++)
33                  {
34                      System.out.print(num);
35
36                      num = (num == 0)? 1 : 0;
37                  }
38              }
39
                System.out.println();
40          }
41
42          sc.close();
43      }
44}
45
46
47
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1010101
0101010
1010101
0101010
1010101
0101010
1010101

Pattern 15 :

1111111
1111122
1111333
1114444
1155555
1666666
7777777

Java Program :

?

```
1 import java.util.Scanner;
2
3 public class MainClass
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8
9         System.out.println("How many rows you want in this pattern?");
10
11        int rows = sc.nextInt();
12
13        System.out.println("Here is your pattern....!!!");
14
15        for (int i = 1; i <= rows; i++)
16        {
17            for (int j = 1; j <= rows-i; j++)
18            {
19                System.out.print(1);
            }
```

```
20
21              for (int j = 1; j <= i; j++)
22              {
23                  System.out.print(i);
24              }
25
26              System.out.println();
27          }
28
29          sc.close();
30      }
31  }
32
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
1111111
1111122
1111333
1114444
1155555
1666666
7777777

Pattern 16 :

0000000
0100000
0020000
0003000
0000400

0000050
0000006

Java Program :

?

```
1  import java.util.Scanner;
2
3  public class MainClass
   {
4      public static void main(String[] args)
5      {
6          Scanner sc = new Scanner(System.in);
7
8          System.out.println("How many rows you want in this pattern?");
9
10         int rows = sc.nextInt();
11
12         System.out.println("Here is your pattern....!!!");
13
14         for (int i = 0; i < rows; i++)
15         {
16             for (int j = 0; j < rows; j++)
17             {
18                 if(i == j)
19                 {
20                     System.out.print(i);
21                 }
                   else
```

```
22                  {
23                      System.out.print(0);
24                  }
25              }
26
27              System.out.println();
28          }
29
30          sc.close();
31      }
32  }
33
34
```

Output :

How many rows you want in this pattern?
7
Here is your pattern….!!!
0000000
0100000
0020000
0003000
0000400
0000050
0000006

Pattern 17 :

1
2 6
3 7 10
4 8 11 13
5 9 12 14 15

Java Program :

?

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.println("How many rows you want in this pattern?");

        int rows = sc.nextInt();

        System.out.println("Here is your pattern....!!!");

        for (int i = 1; i <= rows; i++)
        {
            int num = i;

            for (int j = 1; j <= i; j++)
            {
                System.out.print(num+" ");

                num = num+rows-j;
            }
```

```
24
25              System.out.println();
26          }
27
28          sc.close();
29      }
30  }
31
```

Output :

How many rows you want in this pattern?
5
Here is your pattern….!!!
1
2 6
3 7 10
4 8 11 13
5 9 12 14 15

Pattern 18 :

Different Pyramid Pattern Programs In Java

Pattern 19 :

Diamond Pattern Programs In Java

Pattern 20 :

Floyd’s Triangle In Java

Diamond Of Stars	Diamond Of Numbers
*	1
* *	2 2
* * *	3 3 3
* * * *	4 4 4 4
* * *	3 3 3
* *	2 2
*	1

Java Program To Print Diamond Of Stars(*) :

?

```
1 import java.util.Scanner;
2
3 public class MainClass
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8
```

```
9         //Taking noOfRows value from the user
10
11        System.out.println("How Many Rows You Want In Your Diamond?");
12
13        int noOfRows = sc.nextInt();
14
15        //Getting midRow of the diamond
16
17        int midRow = (noOfRows)/2;
18
19        //Initializing row with 1
20
21        int row = 1;
22
23        System.out.println("Here Is Your Diamond : ");
24
25        //Printing upper half of the diamond
26
27        for (int i = midRow; i > 0; i--)
28        {
29            //Printing i spaces at the beginning of each row
30
31            for (int j = 1; j <= i; j++)
32            {
33                System.out.print(" ");
            }
```

```
34
35              //Printing j *'s at the end of each row
36
37              for (int j = 1; j <= row; j++)
38              {
39                  System.out.print("* ");
40              }
41
42              System.out.println();
43
44              //Incrementing the row
45
46              row++;
47          }
48
49          //Printing lower half of the diamond
50
51          for (int i = 0; i <= midRow; i++)
52          {
53              //Printing i spaces at the beginning of each row
54
55              for (int j = 1; j <= i; j++)
56              {
57                  System.out.print(" ");
58              }

                //Printing j *'s at the end of each row
```

```
59
60              for (int j = row; j > 0; j--)
61              {
62                      System.out.print("* ");
63              }
64
65              System.out.println();
66
67              //Decrementing the row
68
69              row--;
70          }
71      }
72  }
73
74
75
```

Output :

[?](#)

```
How Many Rows You Want In Your Diamond?
7
Here Is Your Diamond :
   *
  * *
 * * *
```

```
* * * *
 * * *
  * *
   *
```

Java Program To Print Diamond Of Numbers :

?

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking noOfRows value from the user

        System.out.println("How Many Rows You Want In Your Diamond?");

        int noOfRows = sc.nextInt();

        //Getting midRow of the diamond

        int midRow = noOfRows/2;

```

```
18          //Initializing row with 1
19
20          int row = 1;
21
22          System.out.println("Here Is Your Diamond : ");
23
24          //Printing upper half of the diamond
25
26          for (int i = midRow; i > 0; i--)
27          {
28              //Printing i spaces at the beginning of each row
29
30              for (int j = 1; j <= i; j++)
31              {
32                  System.out.print(" ");
33              }
34
35              //Printing row value j times at the end of each row
36
37              for (int j = 1; j <= row; j++)
38              {
39                  System.out.print(row+" ");
40              }
41
42              System.out.println();
                //Incrementing the row
```

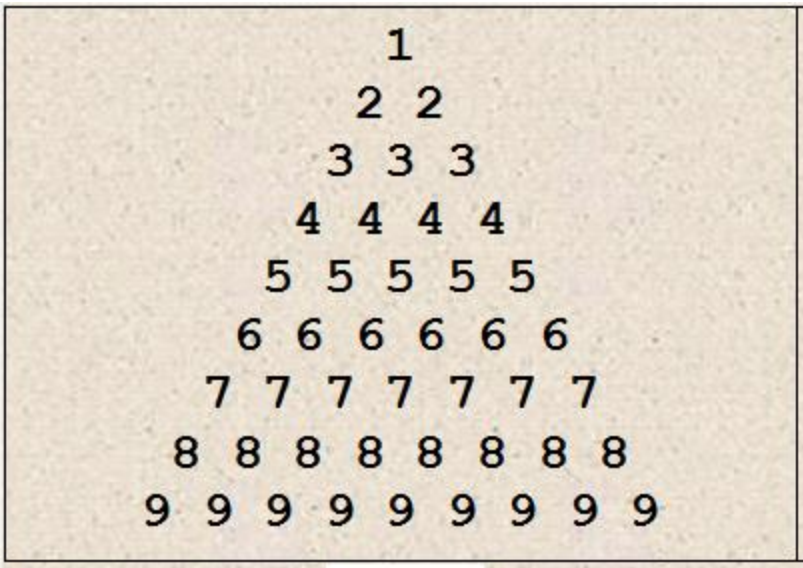
```
43
44              row++;
45          }
46
47          //Printing lower half of the diamond
48
49          for (int i = 0; i <= midRow; i++)
50          {
51              //Printing i spaces at the beginning of each row
52
53              for (int j = 1; j <= i; j++)
54              {
55                  System.out.print(" ");
56              }
57
58              //Printing row value j times at the end of each row
59
60              for (int j = row; j > 0; j--)
61              {
62                  System.out.print(row+" ");
63              }
64
65              System.out.println();
66
67              //Decrementing the row
```

```
68          }
69      }
70  }
71
72
73
74
75
```

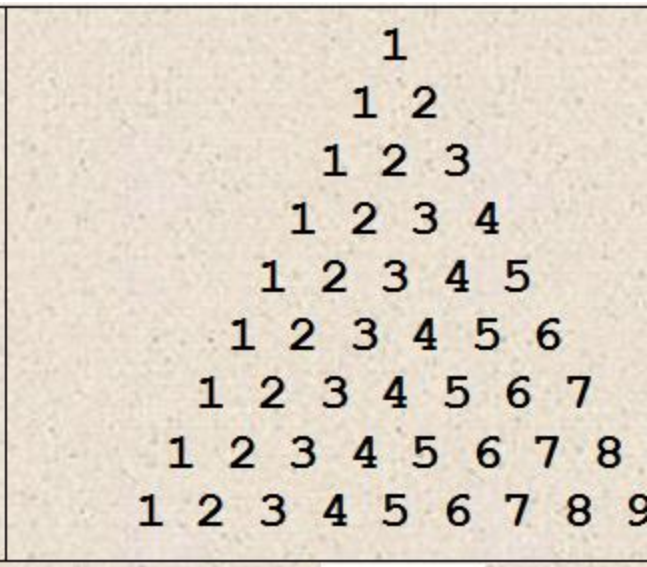
Output :

?

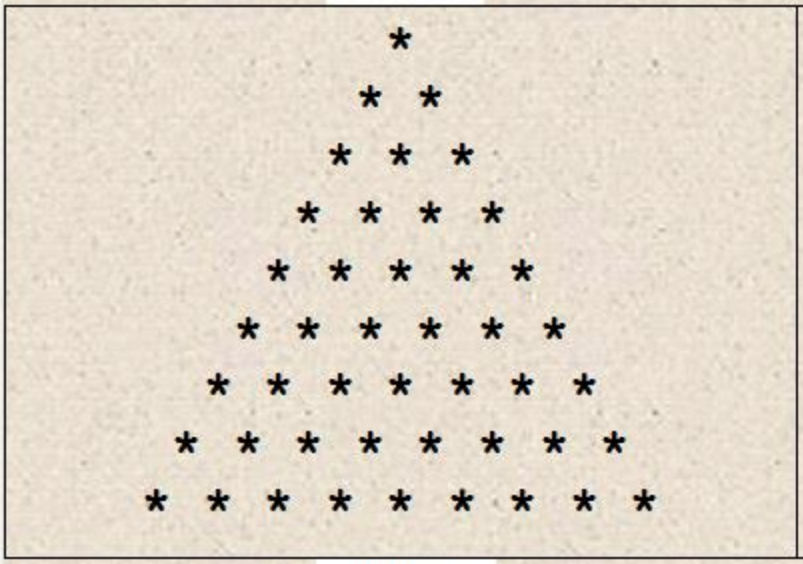
```
1  How Many Rows You Want In Your Diamond?
2  7
3  Here Is Your Diamond :
4      1
5    2 2
6   3 3 3
7  4 4 4 4
8    3 3 3
9      2 2
10       1
```



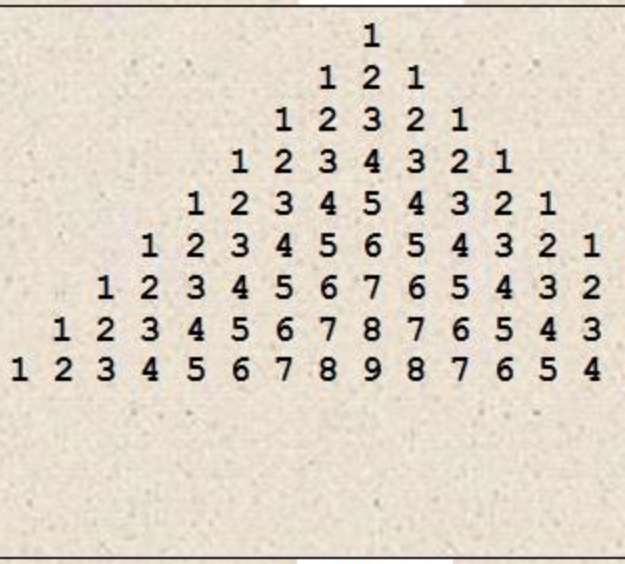
Pattern1



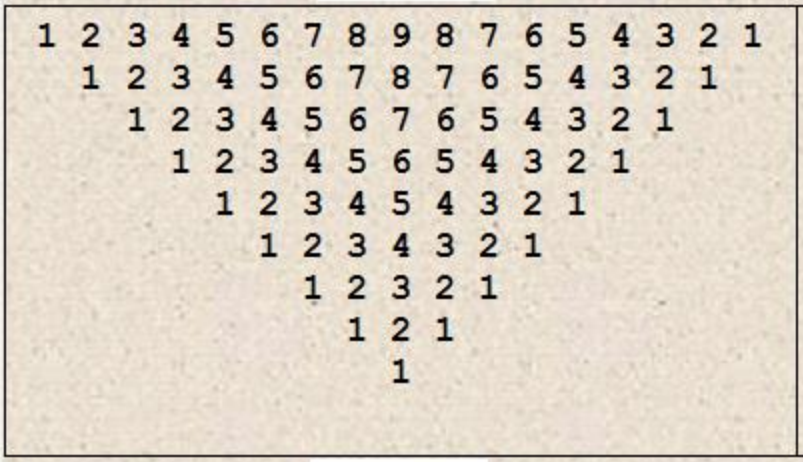
Pattern2



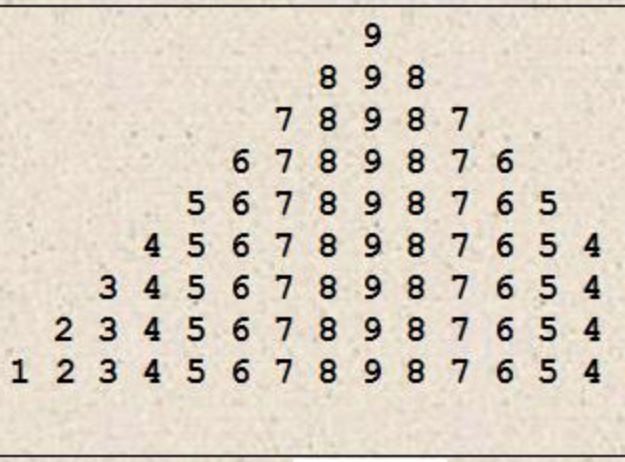
Pattern3



Pattern4



Pattern5



Pattern6

In this post, we will try to write the java programs to create pyramid of numbers in all different patterns.

Pattern 1 : Write java program to create pyramid of numbers like in Pattern1 of the above image?

Take the input from the user and assign it to **noOfRows**. This will be the number of rows he wants in a pyramid. Define one variable called **rowCount** and initialize it to **1.** This will hold the value of current row count. At the beginning of each row, we print **'i'** spaces where **'i'** will be value from **noOfRows** to **1**. At the end of each row, we print **rowCount** value **rowCount** times. i.e in the first row, 1 will be printed once. In the second row, 2 will be printed twice and so on. Below is the java code which implements this logic.

```
?
1  import java.util.Scanner;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
8
9          //Taking noOfRows value from the user
10
11         System.out.println("How Many Rows You Want In Your Pyramid?");
12
13         int noOfRows = sc.nextInt();
14
15         //Initializing rowCount with 1
16
17         int rowCount = 1;
18
19         System.out.println("Here Is Your Pyramid");
20
21         //Implementing the logic
22
23         for (int i = noOfRows; i > 0; i--)
24         {
25             //Printing i spaces at the beginning of each row
26
27             for (int j = 1; j <= i; j++)
28             {
29                 System.out.print(" ");
30             }
31
32             //Printing 'rowCount' value 'rowCount' times at the end of each
row
33
34             for (int j = 1; j <= rowCount; j++)
35             {
36                 System.out.print(rowCount+" ");
```

```
            }

            System.out.println();

            //Incrementing the rowCount

            rowCount++;
        }
    }
}
```

Output :

```
How Many Rows You Want In Your Pyramid?
9
Here Is Your Pyramid
          1
         2 2
        3 3 3
       4 4 4 4
      5 5 5 5 5
     6 6 6 6 6 6
    7 7 7 7 7 7 7
  8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9
```

Pattern 2 : How to create pyramid of numbers in Java like in Pattern2 of the above image?

In this pattern also, we use same logic but instead of printing **rowCount** value **rowCount** times at the end of each row, we print **'j'** where **j** value will be from **1** to **rowCount**.

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking noOfRows value from the user

        System.out.println("How Many Rows You Want In Your Pyramid?");
```

```
        int noOfRows = sc.nextInt();

        //Initializing rowCount with 1

        int rowCount = 1;

        System.out.println("Here Is Your Pyramid");

        //Implementing the logic

        for (int i = noOfRows; i > 0; i--)
        {
            //Printing i spaces at the beginning of each row

            for (int j = 1; j <= i; j++)
            {
                System.out.print(" ");
            }

            //Printing 'j' value at the end of each row

            for (int j = 1; j <= rowCount; j++)
            {
                System.out.print(j+" ");
            }

            System.out.println();

            //Incrementing the rowCount

            rowCount++;
        }
    }
}
```

Output :

```
?
How Many Rows You Want In Your Pyramid?
9
Here Is Your Pyramid
          1
         1 2
        1 2 3
       1 2 3 4
```

```
      1 2 3 4 5
     1 2 3 4 5 6
    1 2 3 4 5 6 7
   1 2 3 4 5 6 7 8
  1 2 3 4 5 6 7 8 9
```

Pattern 3 : Write a java program to create pyramid of stars(*) like in the Pattern3 of the above image?

The same logic is used here also. But, instead of printing **rowCount** or **j** value at the end of each row, we print star(*).

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking noOfRows value from the user

        System.out.println("How Many Rows You Want In Your Pyramid?");

        int noOfRows = sc.nextInt();

        //Initializing rowCount with 1

        int rowCount = 1;

        System.out.println("Here Is Your Pyramid");

        //Implementing the logic

        for (int i = noOfRows; i > 0; i--)
        {
            //Printing i spaces at the beginning of each row

            for (int j = 1; j <= i; j++)
            {
                System.out.print(" ");
            }

            //Printing * at the end of each row

            for (int j = 1; j <= rowCount; j++)
            {
                System.out.print("* ");
```

```
            }

            System.out.println();

            //Incrementing the rowCount

            rowCount++;
        }
    }
}
```

Output :

```
?
1
How Many Rows You Want In Your Pyramid?
9
Here Is Your Pyramid
          *
         * *
        * * *
       * * * *
      * * * * *
     * * * * * *
    * * * * * * *
   * * * * * * * *
  * * * * * * * * *
```

Pattern 4 : Write java program to print pyramid of numbers like in the Pattern4 of the above image?

In this problem, we print **i*2** spaces at the beginning of each row instead of just **i** spaces. At the end of each row, we print **'j'** where **j** value will be from **1** to **rowCount** and from **rowCount-1** to **1**.

```
?
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
```

```
7
8           //Taking noOfRows value from the user
9
10          System.out.println("How Many Rows You Want In Your Pyramid?");
11
12          int noOfRows = sc.nextInt();
13
14          //Initializing rowCount with 1
15
16          int rowCount = 1;
17
18          System.out.println("Here Is Your Pyramid");
19
20          //Implementing the logic
21
22          for (int i = noOfRows; i > 0; i--)
23          {
24              //Printing i*2 spaces at the beginning of each row
25
26              for (int j = 1; j <= i*2; j++)
27              {
28                  System.out.print(" ");
29              }
30
31              //Printing j value where j value will be from 1 to rowCount
32
33              for (int j = 1; j <= rowCount; j++)
34              {
35                  System.out.print(j+" ");
36              }
37
38              //Printing j value where j value will be from rowCount-1 to 1
39
40              for (int j = rowCount-1; j >= 1; j--)
41              {
42                  System.out.print(j+" ");
43              }
44
45              System.out.println();
46
47              //Incrementing the rowCount
48
49              rowCount++;
50          }
51      }
52  }
```

Output :

```
How Many Rows You Want In Your Pyramid?
9
Here Is Your Pyramid
                    1
                  1 2 1
                1 2 3 2 1
              1 2 3 4 3 2 1
            1 2 3 4 5 4 3 2 1
          1 2 3 4 5 6 5 4 3 2 1
        1 2 3 4 5 6 7 6 5 4 3 2 1
      1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
    1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
```

Pattern 5 : Write Java program to print reverse pyramid of numbers like in the Pattern5 of the above image?

In this problem, we iterate outer loop in the reverse order i.e from **1** to **noOfRows** and initialize the **rowCount** to **noOfRows**.

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking noOfRows value from the user

        System.out.println("How Many Rows You Want In Your Pyramid?");

        int noOfRows = sc.nextInt();

        //Initializing rowCount with noOfRows

        int rowCount = noOfRows;

        System.out.println("Here Is Your Pyramid");

        //Implementing the logic

        for (int i = 0; i < noOfRows; i++)
        {
```

```
            //Printing i*2 spaces at the beginning of each row

            for (int j = 1; j <= i*2; j++)
            {
                System.out.print(" ");
            }

            //Printing j where j value will be from 1 to rowCount

            for (int j = 1; j <= rowCount; j++)
            {
                System.out.print(j+" ");
            }

            //Printing j where j value will be from rowCount-1 to 1

            for (int j = rowCount-1; j >= 1; j--)
            {
                System.out.print(j+" ");
            }

            System.out.println();

            //Decrementing the rowCount

            rowCount--;
        }
    }
}
```

Output :

```
?
How Many Rows You Want In Your Pyramid?
9
Here Is Your Pyramid
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
    1 2 3 4 5 6 7 6 5 4 3 2 1
      1 2 3 4 5 6 5 4 3 2 1
        1 2 3 4 5 4 3 2 1
          1 2 3 4 3 2 1
            1 2 3 2 1
              1 2 1
                1
```

Pattern 6 : How do you create pyramid of numbers like in the Pattern6 of the above image?

In this problem, at the end of each row we print **'j'** where **'j'** value will be from **i** to **noOfRows** and from **noOfRows-1** to **i**.

```
import java.util.Scanner;

public class MainClass
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        //Taking noOfRows value from the user

        System.out.println("How Many Rows You Want In Your Pyramid?");

        int noOfRows = sc.nextInt();

        //Initializing rowCount with 1

        int rowCount = 1;

        System.out.println("Here Is Your Pyramid");

        //Implementing the logic

        for (int i = noOfRows; i >= 1; i--)
        {
            //Printing i*2 spaces at the beginning of each row

            for (int j = 1; j <= i*2; j++)
            {
                System.out.print(" ");
            }

            //Printing j where j value will be from i to noOfRows

            for (int j = i; j <= noOfRows; j++)
            {
                System.out.print(j+" ");
            }

            //Printing j where j value will be from noOfRows-1 to i

            for (int j = noOfRows-1; j >= i; j--)
            {
                System.out.print(j+" ");
```

```
38              }
39
40              System.out.println();
41
42              //Incrementing the rowCount
43
44              rowCount++;
45          }
46      }
47  }
48
49
50
51
52
53
```

Input : 5
Output:

```
* * * * *  * * * * *
* * * *      * * * *
* * *          * * *
* *              * *
*                  *
*                  *
* *              * *
* * *          * * *
* * * *      * * * *
* * * * *  * * * * *
```

Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.

This program is divided into four parts.

```
// C program to print

// the given pattern


#include<stdio.h>

pattern(int n)

{

    int i,j;


    // This is upper half of pattern

    for (i=1; i<=n; i++)

    {

        for (j=1; j<=(2*n); j++)

        {

            // Left part of pattern

            if (i>(n-j+1))

                printf(" ");

            else
```

```
            printf("*");

        // Right part of pattern
        if ((i+n)>j)
            printf(" ");
        else
            printf("*");
    }
    printf("\n");
}

// This is lower half of pattern
for (i=1; i<=n; i++)
{
    for (j=1; j<=(2*n); j++)
    {
        // Right Part of pattern
        if (i<j)
            printf(" ");
        else
            printf("*");

        // Left Part of pattern
        if (i<=((2*n)-j))
            printf(" ");
        else
            printf("*");
```

```
        }

        printf("\n");

    }

}


// main function

int main()

{

    pattern(7);

    return 0;

}
```

Output:

```
* * * * * * *  * * * * * * *
* * * * * *      * * * * * *
* * * * *          * * * * *
* * * *              * * * *
* * *                  * * *
* *                      * *
*                          *
*                          *
* *                      * *
* * *                  * * *
* * * *              * * * *
* * * * *          * * * * *
* * * * * *      * * * * * *
* * * * * * * *  * * * * * * *
```

Program to print following pattern:
Examples:

Input : 5
Output:

```
*                   *
* *               * *
* * *           * * *
* * * *       * * * *
* * * * *   * * * * *
* * * * *   * * * * *
* * * *       * * * *
* * *           * * *
* *               * *
```

*   *

Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.

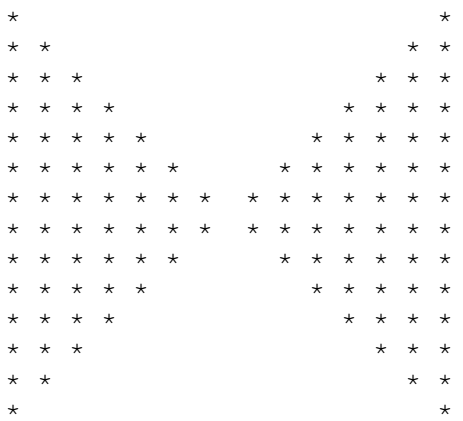
This program is divided into four parts.

```
// C program to print the
// given pattern

#include<stdio.h>
pattern(int n)
{
    int i,j;

    // This is upper half of pattern
    for (i=1; i<=n; i++)
    {
        for (j=1; j<=(2*n); j++)
        {
            // Left part of pattern
            if (i<j)
                printf(" ");
            else
                printf("*");

            // Right part of pattern
            if (i<=((2*n)-j))
                printf(" ");
```

```
            else
                printf("*");
        }
        printf("\n");
    }

    // This is lower half of pattern
    for(i=1; i<=n; i++)
    {
        for(j=1;j<=(2*n);j++)
        {
            // Left part of pattern
            if(i>(n-j+1))
                printf(" ");
            else
                printf("*");

            // Right part of pattern
            if((i+n)>j)
                printf(" ");
            else
                printf("*");
        }
        printf("\n");
    }
}
```

```
// main function

int main()

{

    pattern(7);

    return 0;

}
```

Output:



Program in Java for AMCAT Pattern

```
import java.io.*;
import java.net.*;
public class Selenium_Projext {
public static void main(String[] args){

PrintPat(9);
}

public static void PrintPat(int a)
{ int last=0;
for(int i=1;i<=a;i++)
{ last=last+i;
int l=i;

for(int j=last;l>0;l–)
{

System.out.print(j–);
if(l>1)
System.out.print(“*”);

}

System.out.println();
}
}}
```

```
Input : n = 4
Output :
1
3*2
4*5*6
10*9*8*7

N=5
Output
1
3*2
4*5*6
10*9*8*7
11*12*13*14*15
```

```
1
2 6
3 7 10
4 8 11 13
5 9 12 14 15
```

```
Scanner in = new Scanner(System.in);

System.out.println("How many rows you want in this pattern?");

int rows = in.nextInt();

System.out.println("Here is your pattern….!!!");

for (int i = 1; i <= rows; i++)
{
int num = i;

for (int j = 1; j <= i; j++)
{
System.out.print(num+" ");

num = num+rows-j;
}

System.out.println();
}

in.close();
}
}
```

```
**
****
******
********
```

```java
int k=2;
            for(int i=1;i<=4;i++)
            {
            for(int j=1;j<=k;j++)
            {
            System.out.print("*");
            }
            System.out.println();
            k=k+2;
            }
```

1) Write a java program to find duplicate elements in an array?

First Method : Using Brute Force Method

```
?
1
2  public class MainClass
3  {
4      public static void main(String[] args)
5      {
6          String[] strArray = {"Java", "JSP", "Servlets", "Java", "Struts",
7  "JSP", "JDBC"};
8
9          for (int i = 0; i < strArray.length-1; i++)
10         {
11             for (int j = i+1; j < strArray.length; j++)
12             {
13                 if( (strArray[i].equals(strArray[j])) && (i != j) )
14                 {
15                     System.out.println("Duplicate Element is :
16 "+strArray[j]);
17                 }
18             }
19         }
20 }
21
```

Output :

Duplicate Element is : Java
Duplicate Element is : JSP

Second Method : Using HashSet

```
?
1  import java.util.HashSet;
2
3  public class MainClass
4  {
5      public static void main(String[] args)
6      {
7          String[] strArray = {"Java", "JSP", "Servlets", "Java", "Struts",
8  "JSP", "JDBC"};
9
10         HashSet<String> set = new HashSet<String>();
11
12         for (String arrayElement : strArray)
13         {
14             if(!set.add(arrayElement))
15             {
16                 System.out.println("Duplicate Element is : "+arrayElement);
17             }
```

```
        }
    }
}
```

10) Write a java program to convert an array to ArrayList and an ArrayList to array?

Array To ArrayList :

```
import java.util.ArrayList;
import java.util.Arrays;

public class MainClass
{
    public static void main(String[] args)
    {
        String[] array = new String[] {"ANDROID", "JSP", "JAVA", "STRUTS",
"HADOOP", "JSF"};

        ArrayList<String> list = new
ArrayList<String>(Arrays.asList(array));

        System.out.println(list);
    }
}
```

ArrayList To Array :

```
import java.util.ArrayList;

public class MainClass
{
    public static void main(String[] args)
    {
        ArrayList<String> list = new ArrayList<String>();

        list.add("JAVA");

        list.add("JSP");

        list.add("ANDROID");
```

```
12
13          list.add("STRUTS");
14
15          list.add("HADOOP");
16
17          list.add("JSF");
18
19          String[] array = new String[list.size()];
20
21          list.toArray(array);
22
23          for (String string : array)
24          {
25              System.out.println(string);
26          }
27      }
28  }
29
30
```

12) Write a java program to reverse an array without using an additional array?

```
?
1  import java.util.Arrays;
2
3  public class MainClass
4  {
5      static void reverseArray(int inputArray[])
6      {
7          System.out.println("Array Before Reverse :
"+Arrays.toString(inputArray));
8
9          int temp;
10
11         for (int i = 0; i < inputArray.length/2; i++)
12         {
13             temp = inputArray[i];
14
15             inputArray[i] = inputArray[inputArray.length-1-i];
16
17             inputArray[inputArray.length-1-i] = temp;
          }
```

```
        System.out.println("Array After Reverse :
"+Arrays.toString(inputArray));

        System.out.println("========================================");
    }

    public static void main(String[] args)
    {
        reverseArray(new int[]{4, 5, 8, 9, 10});

        reverseArray(new int[]{12, 9, 21, 17, 33, 7});

        reverseArray(new int[]{891, 569, 921, 187, 343, 476, 555});
    }
}
```