**ANSWER 1:**

import java.util.Scanner;  
  
 public class table {  
 void displaytable(int n){  
 for (int i=1;i<=10;i++){  
 System.*out*.println(n + "\*" + i +"=" + n\*i);  
  
 }  
  
 }  
 public static void main(String[] args){  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("enter any no . to display its table:");  
 int a = in.nextInt();  
 table ob = new table();  
 ob.displaytable(a);  
  
 }  
}

**ANSWER 2:**

public class printjava {  
 static void printj(){  
 for(int i=0;i<4;i++){  
 for (int j=0;j<=3;j++){  
 if(i<=1) {  
 if (j == 3) {  
 System.*out*.print("j");  
 } else {  
 System.*out*.print(" ");  
 }  
 }  
 else if(i==2){  
 if(j==0 || j==3){  
 System.*out*.print("j");  
 }  
 else{  
 System.*out*.print(" ");  
 }  
 }  
 else{  
 if(j==0 || j==3){  
 System.*out*.print(" ");  
 }  
 else{  
 System.*out*.print("j");  
 }  
 }  
  
 }  
 System.*out*.println("");  
 }  
 }static void printA(){  
 for (int i=0;i<4;i++){  
 for (int j=0;j<=4;j++){  
 if(i==0) {  
 if (j == 2) {  
 System.*out*.print("A");  
 } else {  
 System.*out*.print(" ");  
 }  
 }  
 else if(i==1){  
 if(j==1 || j==3){  
 System.*out*.print("A");  
 }  
 else{  
 System.*out*.print(" ");  
 }  
 }  
 else if(i==2){  
 System.*out*.print("A");  
 }  
 else {  
 if (j == 0 || j == 4) {  
 System.*out*.print("A");  
 } else {  
 System.*out*.print(" ");  
 }  
 }  
 }  
 System.*out*.println("");  
 }  
  
 }  
 static void printV()  
  
 {  
 for (int i = 0; i <4; i++) {  
 for (int j = 0; j <= 6; j++) {  
 if(i!=3) {  
 if ((j - i) == 0 || (j + i) == 6) {  
 System.*out*.print("V");  
 } else {  
 System.*out*.print(" ");  
 }  
 }  
 else{  
 if(j==3){  
 System.*out*.print("V");  
 }  
 else{  
 System.*out*.print(" ");  
 }  
 }  
 }  
 System.*out*.println("");  
 }  
 }  
  
  
 public static void main(String[] args){  
 *printj*();  
 *printA*();  
 *printV*();  
 *printA*();  
  
 }  
}

**ANSWER 3:**

import java.util.Scanner;  
public class areaofpolygon {  
  
 public static void main(String[] args) {  
 Scanner in= new Scanner(System.*in*);  
 System.*out*.println("enter ur choice ,1 for area of circle and 2 for any polygon");  
 int i = in.nextInt();  
 if(i==1){  
 System.*out*.println("enter the radius of circle:");  
 double r = in.nextDouble();  
 double area= *circleArea*(r);  
 System.*out*.println("the area of circle is :"+area);  
 }  
 System.*out*.print("Input the number of sides on the polygon: ");  
 int ns = in.nextInt();  
 System.*out*.print("Input the length of one of the sides: ");  
 double side = in.nextDouble();  
 System.*out*.print("The area is: " + *polygonArea*(ns, side)+"\n");  
 }  
 public static double polygonArea(int ns, double side) {  
 return (ns \* (side \* side)) / (4.0 \* Math.*tan*((Math.*PI* / ns)));  
 }  
 public static double circleArea(double r){  
 double ar = Math.*PI*\*r\*r;  
 return(ar);  
 }  
 }

**ANSWER 4:**

public class swapping {  
 int a,b;  
 void swap(int a, int b){  
 int c;  
 this.a= a;  
 this.b=b;  
 System.*out*.println("value of a="+a);  
 System.*out*.println("value of b ="+b);  
 c=this.a;  
 this.a=b;  
 this.b=c;  
 System.*out*.println("after swapping:");  
 System.*out*.println("value of a="+this.a);  
 System.*out*.println("value of b ="+this.b);  
 }  
 public static void main(String[] args){  
 swapping ob = new swapping();  
 ob.swap(2,3);  
  
}  
}

**Answer 5:**

public class decimaltobinary {  
 void A(int n){  
 String bin = Integer.*toBinaryString*(n);  
 System.*out*.println(bin);  
 }  
 void A(String s){  
 int decimal=Integer.*parseInt*(s,2);  
 System.*out*.println(decimal);  
  
 }  
 public static void main( String args[] ) {  
 int dec = 25;  
 String s="11001";  
 decimaltobinary ob = new decimaltobinary();  
 ob.A(dec);  
 ob.A(s);  
  
  
 }  
}

**Answer 6:**

public class binarytodecimal {  
 static String *s* ="10001";  
 static {  
 int decimal = Integer.*parseInt*(*s*, 2);  
 System.*out*.println(decimal);  
 }  
  
 public static void main(String[] args){  
 }  
  
}

**Answer 7:**

public class sumofdigits {  
 int i,sum=0;  
 {  
 while(this.i!=0){  
 int j = this.i%10;  
 this.sum = this.sum + j;  
 i = i/10;  
 }  
 System.*out*.println("the sum of digits of an integer="+sum);  
 }  
 public static void main(){  
 sumofdigits ob =new sumofdigits();  
 ob.i=323;  
  
 }  
}

**Answer 8:**

public class areaofhexagon {  
 public static void main(String[] args){  
 int side = Integer.*parseInt*(args[0]);  
 System.*out*.println("the area of hexagon iis ="+*arhexagon*(side));  
 }  
 public static double arhexagon(int side){  
 double ar = (6\*side\*side)/(4\* Math.*tan*(Math.*PI*/6));  
 return ar;  
 }  
}

**Answer 9:**

import javax.swing.plaf.synth.SynthTextAreaUI;  
import java.util.Scanner;  
  
public class distancebwtopints {  
 public static void main(String[] args){  
 Scanner in =new Scanner(System.*in*);  
 System.*out*.println("enter the value of x1 and y1:");  
 double x1= in.nextInt();  
 double y1 = in.nextInt();  
 Double obj1 = new Double(x1);  
 Double obj2 = new Double(y1);  
 System.*out*.println("enter the value of x2 and y2:");  
 int x2= in.nextInt();  
 int y2 = in.nextInt();  
 Double obj3 = new Double(x2);  
 Double obj4 = new Double(y2);  
 float r= 6371.01f;  
 double d = r\*Math.*acos*((Math.*sin*(obj1)\*Math.*sin*(obj2)+Math.*cos*(obj1)\*Math.*cos*(obj3)\*Math.*cos*(obj2-obj4)));  
 Double ob4 = new Double(d);  
 System.*out*.println("ditsance between two points on earth ="+obj4);  
  
 }  
}

**Answer 10:**

import java.util.Scanner;  
public class reverseastring {  
 public static void main(String[] args)  
 {  
 System.*out*.println("Enter string to reverse:");  
  
 Scanner read = new Scanner(System.*in*);  
 String str = read.nextLine();  
 String reverse = "";  
  
  
 for(int i = str.length() - 1; i >= 0; i--)  
 {  
 reverse = reverse + str.charAt(i);  
 }  
  
 System.*out*.println("Reversed string is:");  
 System.*out*.println(reverse);  
 }  
 }

**Answer 11:**

import static java.lang.Character.\*;  
  
public class countliterals {  
 public static void main(String[] args){  
 int countletter=0,countdigit=0,countspaces=0;  
 StringBuffer ob = new StringBuffer("hello hi no 4");  
 for(int i=0; i<ob.length();i++){  
 char c = ob.charAt(i);  
 if(*isLetter*(c)){  
 countletter = countletter + 1;  
 }  
 else if(*isDigit*(c)){  
 countdigit = countdigit + 1;  
 }  
 else{  
 countspaces = countspaces +1;  
 }  
 }  
 System.*out*.println("the no of letetrs:"+countletter);  
 System.*out*.println("the no of digit:"+countdigit);  
 System.*out*.println("the no of whitespaces:"+countspaces);  
 }  
}

**Answer 12:**

import java.util.\*;  
  
public class counttokens {  
 public static void main(String args[]) {  
  
 // Creating a StringTokenizer  
 StringTokenizer str\_arr  
 = new StringTokenizer(  
 "hi friends , whats up!");  
  
 // Counting the tokens  
 int count = str\_arr.countTokens();  
 System.*out*.println("Total number of Tokens: " + count);  
 }  
}

**Answer 13:**

import java.util.Scanner;  
public class computeonint {  
 {  
 int n;  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("enter any no.:");  
 n = in.nextInt();  
 int j = n\*10+n;  
 int x = n\*100 + n\*10 + n;  
 int sum = n+j+x;  
 System.*out*.println("the sum of "+n+"+"+j+"+"+x+"="+sum);  
 }  
 public static void main(String[] args){  
 computeonint ob = new computeonint();  
  
 }  
}

**Answer 14:**

import java.util.Scanner;  
public class evenorodd {  
 public static void main(String[] args){  
 Scanner in = new Scanner(System.*in*);  
 int n = in.nextInt();  
 if(n%2==0){  
 System.*out*.println(0);  
 }  
 else{  
 System.*out*.println(1);  
 }  
 }  
}

**Answer 15:**

public class stringtoint {  
 public static void main(String[] args){  
 String number = "10";  
 int result = Integer.*parseInt*(number);  
 System.*out*.println(result);  
  
 }  
}

**Answer 16:**

public class capitalisfirstword {  
 static String convert(String str)  
 { char ch[] = str.toCharArray();  
 for (int i = 0; i < str.length(); i++) {  
 if (i == 0 && ch[i] != ' ' ||  
 ch[i] != ' ' && ch[i - 1] == ' ') {  
 if (ch[i] >= 'a' && ch[i] <= 'z') {  
 ch[i] = (char)(ch[i] - 'a' + 'A');  
 }  
 }  
 else if (ch[i] >= 'A' && ch[i] <= 'Z')  
 ch[i] = (char)(ch[i] + 'a' - 'A');  
 }  
 String st = new String(ch);  
 return st;  
 }  
  
 public static void main(String[] args)  
 {  
 String str = "hii cet";  
 System.*out*.println(*convert*(str));  
 }  
}

**Answer 17:**

public class stringtolowercase {  
 public static void main(String args[])  
 {  
 String s = "Welcome! to Cet.";  
 String s1 = s.toLowerCase();  
 System.*out*.println(s1);  
 }  
}

**Answer 18:**

import java.util.Scanner;  
public class reverseaword {  
 public static void main(String[] args){  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.print("\nInput a word: ");  
 String word = in.nextLine();  
 word = word.trim();  
 String result = "";  
 char[] ch=word.toCharArray();  
 for (int i = ch.length - 1; i >= 0; i--) {  
 result += ch[i];  
 }  
 System.*out*.println("Reverse word: "+result.trim());  
 }  
}

**Answer 19:**

public class insertofword {  
 public static void main(String[] args)  
 {  
 String main\_string = "welcome cet";  
 String word = "T0";  
 System.*out*.println(main\_string.substring(0, 82) + word + main\_string.substring(7));  
 }  
}

**Answer 20:**

public class runtime {  
 public static void main(String[] args){  
 long time1 = System.*nanoTime*();  
 System.*out*.println("hello, check check");  
 long time2 = System.*nanoTime*();  
 long timeTaken = time2 - time1;  
 System.*out*.println("Time taken " + timeTaken + " ns");  
 }  
}

**Answer 21:**

import java.util.Scanner;  
public class oddeven {  
 public static void main(String[] args){  
 int[] a = new int[6];  
 Scanner ob = new Scanner(System.*in*);  
 for(int i=0;i<6;i++){  
 a[i]= ob.nextInt();  
 }  
 int[] b = new int[6];  
 int count=0;  
 for (int j=0;j<6;j++){  
 if(a[j]%2!=0){  
 b[count]=a[j];  
 count++;  
 }  
 }  
 for (int j=0;j<6;j++) {  
 if (a[j] % 2 == 0) {  
 b[count] = a[j];  
 count++;  
 }  
 }  
 for(int j=0;j<6;j++){  
 System.*out*.println(b[j]);  
 }  
  
  
 }  
}

**Answer 22:**

public class numberformaterror  
{  
 public static void main(String args[])  
 {  
 String str1= "10";  
 int x = Integer.*parseInt*(str1);  
 System.*out*.println(x\*x); // prints 100  
  
 try  
 {  
 String str2= "ten";  
 int y = Integer.*parseInt*(str2);  
 }  
 catch(NumberFormatException e)  
 {  
 System.*err*.println("Unable to format. " + e);  
 }  
 }  
}

**Answer 23:**

public class fatorial {  
 public static void main(String[] args){  
 int n=5;  
 int fact=1;  
 for(int i=1;i<=5;i++){  
 fact=fact\*i;  
 }  
 int count=0;  
 while(fact!=0){  
 int j = fact%10;  
 if(j==0){  
 count++;  
 }  
 fact=fact/10;  
 }  
 System.*out*.println(count);  
 }  
  
}

**Answer 24:**

import java.util.Scanner;  
public class palindrome {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 int num = in.nextInt();  
 int reversedInteger = 0, remainder, originalInteger;  
 originalInteger = num;  
 while( num != 0 )  
 {  
 remainder = num % 10;  
 reversedInteger = reversedInteger \* 10 + remainder;  
 num /= 10;  
 }  
 if (originalInteger == reversedInteger)  
 System.*out*.println(originalInteger + " is a palindrome.");  
 else  
 System.*out*.println(originalInteger + " is not a palindrome.");  
 }  
}

**Answer 25:**

import java.util.\*;  
  
public class median  
{  
 public static double findMedian(int a[], int n)  
 {  
 Arrays.*sort*(a);  
 if (n % 2 != 0)  
 return (double)a[n / 2];  
  
 return (double)(a[(n - 1) / 2] + a[n / 2]) / 2.0;  
 }  
  
 public static void main(String args[])  
 {  
 int a[] = { 1, 3, 4, 2, 7, 5, 8, 6 };  
 int n = a.length;  
 System.*out*.println("Median = " + *findMedian*(a, n));  
 }  
}

**Answer 26:**

class duplicate  
{ static int removeDuplicates(int arr[], int n)  
 {  
 if (n==0 || n==1)  
 return n;  
  
 int[] temp = new int[n];  
 int j = 0;  
 for (int i=0; i<n-1; i++)  
 if (arr[i] != arr[i+1])  
 temp[j++] = arr[i];  
 temp[j++] = arr[n-1];  
  
 for (int i=0; i<j; i++)  
 arr[i] = temp[i];  
  
 return j;  
 }  
  
 public static void main (String[] args)  
 {  
 int arr[] = {1, 2, 2, 3, 4, 4, 4, 5, 5};  
 int n = arr.length;  
  
 n = *removeDuplicates*(arr, n);  
 for (int i=0; i<n; i++)  
 System.*out*.print(arr[i]+" ");  
 }  
}

**Answer 27:**

**Autoboxing:**Converting a primitive value into an object of the corresponding [wrapper class](https://www.geeksforgeeks.org/wrapper-classes-java/) is called autoboxing.

**For ex:**

import java.io.\*;

class autoboxing

{

    public static void main (String[] args)

    {

      byte i1 = 1;

Byte i = new Byte(i1);

        System.out.println("Value of i: " + i);

        System.out.println("Value of i1: " + i1);

}

}

**Unboxing:** Converting an object of a wrapper type to its corresponding primitive value is called unboxing.

**For ex:**

import java.io.\*;

class autoboxing

{

    public static void main (String[] args)

    {

        Byte i = new Byte(1);

        byte i1 = i;

        System.out.println("Value of i: " + i);

        System.out.println("Value of i1: " + i1);

}

}

**Answer 28:**

**Jvm: JVM** *[Java Virtual Machine]* is a part of JRE(Java Run Environment) that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language.

Functions of jvm are :

1. Loading bytecodes from the class files  
    2. Verifying the loaded byte codes  
    3. Linking the program with the necessary libraries  
    4. Memory Management by Garbage Collection  
    5. Managing calls between the program and the host environment.

**Answer 29:**

Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Thus when you write a piece of Java code in a particular platform and generated an executable code .class file. You can execute/run this .class file on any system the only condition is that the target system should have JVM (JRE) installed in it.

In short, Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

**Answer 30:**

1) **Own Memory Management**

The memory management mechanism is unique and is owned by Java. There is no need for manual intervention for garbage collection and everything is handled automatically. There is no need for a headache to free the memories. It drastically reduces the programmer overhead. Therefore the programmer’s hand must be free from memory management. Relieving the memory in Java is the job of JVM.

2) **No Pointers**

There is no support for the pointers concept in Java. Some of the arbitrary memory locations can be addressed with the help of pointers for doing read and write operations which are unauthorized. This does not serve the purpose of being secured. That is why users do not use the concept of pointers.

3) **Access Specific Keywords**

There is another benefit for Java security and it is nothing but having an access specific keyword. If access to a method is what programmer wants to give to other functions then the public keyword must be used. If he/she wants to hide the information then private keywords must be used. Access security issues can be controlled by the programmer like the above-mentioned line. For avoiding data to be overridden the programs can use the final keyword.

Java: The Most Secure Programming Language?

4) **Compile Time Checking**

It is more secure because of the ability of compile-time checking. For instance, if a method which is unauthorized and wanting to access a variable which private then the JVM will catch the error during compile time. For avoiding system crash JVM catches lots of errors which is actually a great thing. Two different results are produced with the help of two different Java compilers. Memory locations cannot be accessed other than the array as it checks array bounds carefully.

**Answer 31:**

**1) Checked:** are the exceptions that are checked at compile time. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using throws keyword.

For example, consider the following Java program that opens file at location “C:\test\a.txt” and prints the first three lines of it. The program doesn’t compile, because the function main() uses FileReader() and FileReader() throws a checked exception FileNotFoundException. It also uses readLine() and close() methods, and these methods also throw checked exception IOException.

import java.io.\*;

class Main {

    public static void main(String[] args) {

        FileReader file = new FileReader("C:\\test\\a.txt");

        BufferedReader fileInput = new BufferedReader(file);

            for (int counter = 0; counter < 3; counter++)

            System.out.println(fileInput.readLine());

        fileInput.close();

    }

}

**Unchecked** are the exceptions that are not checked at compiled time. In C++, all exceptions are unchecked, so it is not forced by the compiler to either handle or specify the exception. It is up to the programmers to be civilized, and specify or catch the exceptions.  
In Java exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under throwable is checked.

Eg:

class Main {

   public static void main(String args[]) {

      int x = 0;

      int y = 10;

      int z = y/x;

  }

}

**Answer 32:**

1. Object
2. Class
3. Data Hiding and Encapsulation
4. Dynamic Binding
5. Message Passing
6. Inheritance
7. Polymorphism

**OBJECT:** Object is a collection of number of entities. Objects take up space in the memory. Objects are instances of classes. When a program is executed , the objects interact by sending messages to one another. Each object contain data and code to manipulate the data. Objects can interact without having know details of each others data or code.

**CLASS:**Class is a collection of objects of similar type. Objects are variables of the type class. Once a class has been defined, we can create any number of objects belonging to that class. Eg: grapes bannans and orange are the member of class fruit.  
*Example:*

Fruit orange;

In the above statement object mango is created which belong to the class fruit.

**DATA ABSTRACTION AND ENCAPSULATION:**

Combining data and functions into a single unit called **class** and the process is known as **Encapsulation**.Data encapsulation is important feature of a class. Class contains both data and functions. Data is not accessible from the outside world and only those function which are present in the class can access the data. The insulation of the data from direct access by the program is called data hiding or information hiding. Hiding the complexity of proram is called **Abstraction** .

**DYNAMIC BINDING:** Refers to linking of function call with function defination is called binding and when it is take place at run time called dynamic binding.

**MESSAGE PASSING:** The process by which one object can interact with other object is called message passing.

**INHERITANCE:** it is the process by which object of one class aquire the properties or features of objects of another class. The concept of inheritance provide the idea of reusability means we can add additional features to an existing class without Modifying it. This is possible by driving a new class from the existing one. The new class will have the combined features of both the classes.  
*Example:* **Robine** is a part of the class flying bird which is again a part of the class bird.

**POLYMORPHISM:** A greek term means ability to take more than one form. An operation may exhibite different behaviours in different instances. The behaviour depends upon the types of data used in the operation.

*Example:*

* Operator Overloading
* Function Overloading

.