# B1.

The features of java are as given below :- 1.simple

2.Object oriented 3.robust

4.plateform independent 5.secure

1. multi threading 7.high performance

# B2.

JVM is Java Virtual Machine is an abstract machine. It does not exist physically. It provides a runtime environment in which java bytecode is executed. It also executes those codes which are written in other languages and compiled to Java bytecode. JVM loads code, verify codes, run codes, and provides a runtime environment.

JRE is Java Runtime Environment. It is an implementation of JVM. It physically exists. It contains sets of libraries and other files that JVM uses at runtime.

JDK is Java Development Kit that provides the software development environment which is used to develop java applications and applets. It contains JRE and other development tools. It contains privates JVM and a few other resources such as interpreter/loader, compiler, document generator, etc to complete the development of Java application.

# B3.

Unlike other programming languages such as C, C++ etc which are compiled into

platform-specific machines. Java is guaranteed to be a write-once, run-anywhere language.

On compilation, Java program is compiled into bytecode. This bytecode is

platform-independent and can be run on any machine, plus this bytecode format also provides security. Any machine with Java Runtime Environment can run Java Programs.

# B4.

The three flavors of java are:-

* 1. Java standard edition[JAVA SE]
  2. Java micro edition[JAVA ME]
  3. Java enterprise edition[JAVA EE]

# B5.

There are two types of memories in the java virtual machine. They are:

1. Heap Memory
2. Stack memory

# B6.

The latest version of java is 17.

# B7.

Unlike other programming languages such as C, C++, etc which are compiled into

platform-specific machines. Java is guaranteed to be a write-once, run-anywhere language.

# B8.

No java isn’t the fully object-oriented language. Pure Object-Oriented Language or Complete Object-Oriented Language are Fully Object-Oriented Language which supports or have features that treats everything inside program as objects. It doesn’t support primitive datatype(like int, char, float, bool, etc.). There are seven qualities to be satisfied for a programming language to be pure Object Oriented are abstraction, encapsulation, etc.

# B9.

Bytecode is the intermediate representation of a Java program, allowing a JVM to translate a program into machine-level assembly language. When a Java program is compiled, bytecode is generated in the form of a *.class* file. This *.class* file contains non-runnable instructions and relies on a JVM to be interpreted.

# B10.

Heap Memory is the memory where the object resides.

# B11.

JAR file stands for Java Archive while WAR file stands for Web Application Resource or Web Application Archive, and EAR stands for Enterprise Application Archive.A JAR file is a file with Java classes, associated metadata, and resources such as text and images aggregated into one file. A WAR file is a file that is used to distribute a collection of JAR files, JSP, Servlet, XML files, static web pages like HTML and other resources that constitute a web application. An EAR file is a standard JAR file that represents the modules of the application, and a metadata directory called META-INT which contains one or more deployment descriptors. Thus, this is the main difference between JAR WAR and EAR files.

# B12.

A Memory Leak is a situation when there are objects present in the heap that are no longer used, but the garbage collector is unable to remove them from memory and, thus they are

unnecessarily maintained. A memory leak is bad because it blocks memory resources and degrades system performance over time.

# B13.

Java garbage collection is an automatic process. Automatic garbage collection is the process of looking at heap memory, identifying which objects are in use and which are not, and deleting the unused objects. An in-use object, or a referenced object, means that some part of your program still maintains a pointer to that object. An unused or unreferenced object is no longer referenced by any part of your program. So the memory used by an unreferenced object can be reclaimed. The programmer does not need to mark objects to be deleted explicitly. The garbage collection implementation lives in the JVM.

# B14.

Garbage collector sweeps heap memory only. Usually, stack memory is collected automatically when the execution path reaches the end of the scope.

# B15.

It is the task of garbage collection (GC) in the Java virtual machine (JVM) to automatically determine what memory is no longer being used by a Java application and to recycle this memory for other uses. Because unreferenced objects are automatically removed from the heap memory, GC makes Java memory-efficient.

# B16.

public class Main {

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

System.*out*.println("input number a= "); int a=sc.nextInt();

System.*out*.println("input number b= "); int b=sc.nextInt();

System.*out*.println("input number c= "); int c=sc.nextInt();

if(a>b&&a>c){

System.*out*.println("a = "+a+" is the greatest number.");

}else if(b>a&&b>c){

System.*out*.println("b = "+b+" is the greatest number.");

}else{

System.*out*.println("c = "+c+" is the greatest number.");

}

}

}

# B18.

public class Main {

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

System.*out*.println("Input the year= "); int a=sc.nextInt();

if(a%4==0){

System.*out*.println("The "+ a+" is a Leap year");

}else

System.*out*.println("This "+ a+" is not a leap year");

}

}

# B19.

public class Main {

public static void main(String[] args) {

//Scanner sc = new Scanner(System.in); System.*out*.println("the first 10 natural numbers are ="); int i=1;

while(i<=10){ System.*out*.println(i); i++;

}}}

# B20.

public class Main {

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

int sum =0,avg=0;

System.*out*.println("Input the 5 integer values "); for(int i=1;i<=5;i++){

int a=sc.nextInt(); sum=sum+a; avg=sum/i;

}

System.*out*.println("The sum= "+ sum+"\nThe average = "+avg);

}}

# B21.

**class Main {**

**public static void main(String[] args) { int n=4;**

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

**System.*out*.print( j);**

**}**

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

**}}}**

# B22.

**class Main {**

**public static void main(String[] args) { int num=1, n=4;**

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

**System.*out*.print(num+" "); num++;**

**}**

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

**}}}**

# B23.

**class Main {**

**public static void main(String[] args) { int n=5;**

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

**if((i+j)%2==0){**

**System.*out*.print(1);**

**}else**

**System.*out*.print(0);**

**}}}**

**}**

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

# B24.

**class Main {**

**public static void main(String[] args) { int n=4;**

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

**int num= (int) Math.*pow*(11,i); if(i>2){**

**System.*out*.print(num); break;**

**}**

**else**

**System.*out*.print(1);**

**}**

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

**}}}**

# B25.

**public class Main {**

**public static void main(String[] args) { Scanner sc = new Scanner(System.*in*); System.*out*.println("Input the number= "); String a=sc.next();**

**int s=Integer.*valueOf*(a);**

**System.*out*.println("The digits in the given integer number ="+a.length());**

**}}**

# B26.

**public class Main {**

**public static void main(String[] args) { int letters=0;**

**int numbers=0; int other=0;**

**int space=0;**

**Scanner sc = new Scanner(System.*in*); System.*out*.println("Input the String= "); String a = sc.nextLine();**

**char[] c= a.toCharArray(); for(int i=0;i<a.length();i++){**

**if(Character.*isLetter*(c[i])){ letters++;**

**}**

**else if(Character.*isDigit*(c[i])){ numbers++;**

**}else if(Character.*isSpaceChar*(c[i])){ space++;**

**} else {**

**other++;**

**}**

**}**

**System.*out*.println("letters ="+letters+" numbers= "+numbers+" space**

**="+space+" other Characters = "+other);**

**}}**

# B27.

**public class Main {**

**public static void main(String[] args) { Scanner sc = new Scanner(System.*in*); System.*out*.println("Input the character= "); char a = sc.next().charAt(0);**

**int b=a;**

**System.*out*.println( "The ASCII value of the given character is"+b);**

**}}**

# B28.

**public class Main {**

**public static void main(String[] args) { Scanner sc = new Scanner(System.*in*); System.*out*.println("Input the integer = "); int b=sc.nextInt();**

**String a=String.*valueOf*(b); String i =a.concat(a);**

**System.*out*.println(a +"+"+i+"+"+i.concat(a));**

**}}**

# B29.

**public class Main {**

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

**SimpleDateFormat ftr=new SimpleDateFormat("HH:mm:ss dd/MM/yyyy"); Date d=new Date();**

**System.*out*.println("THE System time is = "+ftr.format(d));**

**}}**

# B30.

**public class Main {**

**public static void main(String[] args) { int[] a=new int[101];**

**for(int i=1;i<101;i++){ a[i]=i;**

**}**

**System.*out*.println("number Divisible by 3:"); *Divisible*(a,3,3,a.length-1); System.*out*.println("\nnumber Divisible by 5:"); *Divisible*(a,5,5,a.length-1); System.*out*.println("\nnumber Divisible by 3 and 5:"); *Divisible*(a,3,5,a.length-1);**

**}**

**public static void Divisible(int Array[],int n1,int n2 ,int i){**

**for(i=1;i< Array.length;i++){ if(Array[i]%n1==0&&Array[i]%n2==0)**

**System.*out*.print(Array[i]+" ");**

**}**

**}**

**}**

# B31.

**public class Main {**

**public static void main(String[] args) { boolean isSum=true;**

**Scanner s = new Scanner(System.*in*); System.*out*.println("input first Value :"); int a=s.nextInt();**

**System.*out*.println("input Second Value :"); int b=s.nextInt();**

**System.*out*.println("input third Value :"); int c=s.nextInt();**

**if(a+b==c){ System.*out*.println(isSum);**

**}else**

**System.*out*.println(!isSum);**

**}**

**}**

# B32.

**public class Main {**

**public static void main(String[] args) { Scanner s = new Scanner(System.*in*); System.*out*.println("input first Value :"); int a=s.nextInt();**

**System.*out*.println("input Second Value :"); int b=s.nextInt(); System.*out*.println("result: "+*isSame*(a,b));**

**}**

**public static boolean isSame(int i,int j){ if(i<25||i>75||j<25||j>75){**

**return false;**

**}**

**int a1=i%10; int b1=j%10; i/=10;**

**j/=10; return(a1==b1||a1==j||b1==i||i==j);**

**}}**

# B33.

**public class Main {**

**public static void main(String[] args) { int sum=0,j=0,count;**

**for(int i=1;i<=100;i++) { count=0; for(j=2;j<=i/2;j++){**

**if(i%j==0){ count++; break;**

**}**

**}**

**if(count==0&&i!=1){ sum=sum+i;**

**}**

**}**

**System.*out*.println("Sum of prime numbers are: "+sum);**

**}**

**}**

# B34.

**public class Main {**

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

**Scanner sc=new Scanner(System.*in*); System.*out*.println("The first Double value is: "); double i=sc.nextDouble();**

**System.*out*.println("The Second Double value is: "); double j=sc.nextDouble();**

**sum=i+j;**

**System.*out*.println("Sum of the numbers is between 0 and 1 : "); if(sum>1||sum<0)**

**System.*out*.println(false); else**

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

**}**

**}**

# B35.

**public class Main {**

**public static void main(String[] args) { int sum,num;**

**Scanner sc=new Scanner(System.*in*); System.*out*.println("Input 6 digit non-negative number: "); int i=sc.nextInt();**

**Stack s=new Stack<>();**

**System.*out*.println("the number after Splitting : "); while(i>0) {**

**// System.out.print(i%10+" "); s.push(i%10);**

**i=i/10;**

**}**

**while(!s.isEmpty()) System.*out*.print(s.pop()+" ");**

**}**

**}**