CEP-01

Q.1 Explain Features of Java.

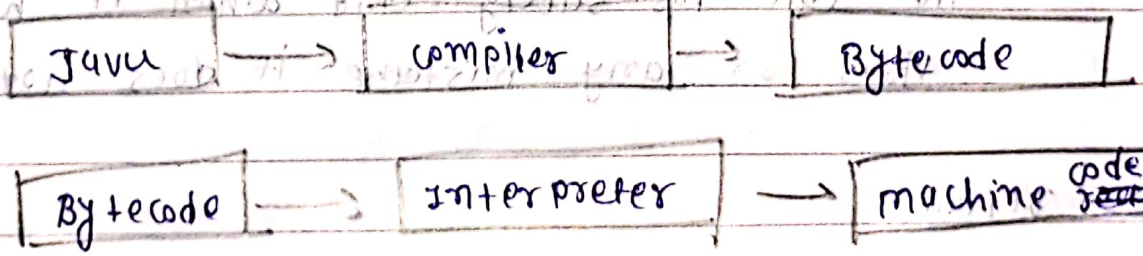
Ans → There are many features of Java:

① Object oriented

→ In this all data are in class or object and it supports the object oriented programming and it also supports the inhetents and polymofil.

② Complied & Interpreted

→ complied or Interpreted are used for the convert the lunguge from high level to low level but Java used both for transform high level to low level



③ platform Independent & portable.

→ Java program is platform independent because once you write the code and complie the program

then you can run this program on any platform like (windows, Linux, macos etc). It is not depend dependent on machine.

(4) Distributed :-

→ This features is cellled that in Java program is easily run in the network. So we can develop the program or application which can easily run on the Network and for this Java have java.net Packege

(5) Robust & secure

→ It have the accession handling which identify the any error. Before the execution the program on any platform it cheks the code for safety and when it have virus or any mistake it does not execute

Q=2 Define :-

Ans ⇒ (1) Byte code :-

Java bytecode is the instruction set for the java virtual machine. It act similar to an assembler which is an alias representation of a C++ code. As soon as a Java program is compiled

માણસને પૂર્વગ્રહ સિવાય બીજો કોઈ ગ્રહ નડતો નથી.

Java bytecode is generated

(2) JVM :-

The Java virtual machine (JVM) is the runtime engine of the Java platform, which allows any program written in Java or other language compiled into Java bytecode to run on any computer that has a native JVM.

(3) JDK :-

The Java development kit (JDK) is a software development environment used for developing Java applications and applets. it includes the Java Runtime environment (JRE), an interpeter/ loader (Java) and other tools in java development.

(4) JIT :-

JIT (Just-in time complier) is a complier that converts Program source code into native machine code just before the program is run.

In case of Java JIT complier converts Java's intermediate language into native machine code as needed

(5) unicode :-

Unicode is a 16-bit Character encoding standard and is capable to represent almost every character

of well-known languages of the world.

→ Before unicode, there were multiple standards to represent character encoding:

1) ASCII - for the United states
2) ISO 8859-1 for western European language
3) KOI-8 for Russian
4) GB18030 and BIG-5 for chinese.

⑥ Short-circuit operators :-

Boolean operators And & OR are called short circuit logical operators in Java.

AND :- If first value is false second value is not evaluated. {&&}

OR :- If first value is true the second value is not evaluated.

ex:

```
      false                 Not evaluated
if ((0 == 1)  &&  (2 == 2))
    S.O.P ("A");   ✗

      true                  not evaluated
if ((1 == 1)  ||  (2 == 3))
    S.O.P ("B");   ✓
```

Q-3 compare object oriented Programming with sequential Programming.

Any 2)

Sequential ~~oriented~~ programming (Procedure oriented Programming)	object oriented programming (OOP)
① Priority is on doing things not on data, means it is function driven.	① Priority is on data rather than procedure, means object drivien.
② Top down approach in Program design	② Bottom up approach in Program design.
③ Large Programs are divided into smaller Programs known as functions.	③ large Programs are divided into classes and object.
④ Adding of data and function is difficult.	④ Adding of data and function is easy.

પ્રશ્નને પ્રાર્થના એ શક્તિશાળી હથીયાર છે.

(5) we cannot declare namespace directly	(5) we can use name spure directly
(6) ex:- C, Fortan, Pascal, etc..	(6) ex:- C++, Java, C#, etc..

Q=4 Explain command line argument with help of example.

Ans 3) There may be the situation where you will want to pass information into a program when you run it. This is accomplished by passing command-line arguments to main().

→ A command-line arguments is the information that directly follows the program's name on the command line when it is executed. They are stored as strings in a string array passed to the args parameter of main(). So, accessing the command-line arguments inside a Java Program is quite easy.

→ The first command-line argument is stored at args[0], the second at args[1], and so on

Ex:-
```
class commandLine {
public static void main (String args[]) {
```

```
for (int i=0; i<args.length; i++)
    System.out.println("args[" + i + "]: " + args[i]);
  }
}
```

→ By executing this program:-

Java commandLine this is a test 100 -1

→ Output

args[0] : this
args[1] : is
args[2] : a
args[3] : test
args[4] : 100
args[5] : -1

Q=5 (a) write a program for fibonacci series in Java.

```
class fibonacciExample1 {
  public static void main (String args[])
{
    int n1=0, n2=1, n3, i, count=10;
    System.out.print(n1+" "+n2);

    for (i=2; i<count; ++i) // loop starts from 2 because 0 and
                            1 are already printed.
```

```
{
    n3 = n1 + n2;
    System.out.println(" " + n3);
    n1 = n2;
    n2 = n3;
  }
}
```

→ Output:

0 1 1 2 3 5 8 13 21 34

(b) write a program to find the given number is Armstrong or not.

Ans →

```
class Armstrong_Example {
  public static void main (String[] args) {
    int c=0, a, temp;
    int n=153;
    temp=n;
    while (n>0)
    {
      a=n%10;
      n=n/10;
      c = c + (a*a*a);
    }
    if (temp == c)
      System.out.println("armstrong number");
```

```
else
    system.out.println ("not armstrong number");
    }
}
```

→ output:-

Armstrong number.

→ (c) factorial of a given number.

Ans ⇒

```
class factorial Example {
public static void main (string args []) {
int i, fact = 1;

int number = 5
for (i = 1; i <= number; i++) {
    {
      fact = fact * i;
    }
system.out.println ("factorial of" + number + "is:" + fact);
  }
}
```

→ output:-

factorial of 5 is: 120