Q 1 What is difference between C++ and Java ?

C++	Java
• C++ is platform dependent.	• Java is platform independent.
• c++ mainly used for System Programming	• Java is mainly used in application programming. It is widely used in window, web-based, enterprise and mobile applications.
• C++ supports goto statement.	• Java doesn't support goto statement.
• C++ supports operator overloading.	• Java doesn't support operator overloading.
• You can write pointers in C++	• Java support pointer internally. However you can't write pointer in java.
• C++ uses compiler only	• Java uses compiler and interpreter both. Java is interpreted that is why platform-independent.
• C++ call by value and call by refrence	• Java supports call by value only. There is no call by refrence.
• C++ is nearer to hardware	• Java is not so interactive with hardware.
• C++ supports multiple inheritence	• Java doesn't support multiple inheritence through class by can be achieved using interfaces.

Page No.
Date

Q1 What are key features of java?

- Simple - Java is very easy to learn and its syntax is simple & clean Java has removed complicated and rarely used features eg. pointers, operator overloading.

Object-Oriented :- Object-Oriented programming is methodology that simplifies software developement and maintainence by providing some rules. and also use of below concepts.
1. Object 2. Class 3. Inheritence 4. Polymorphism
5. Abstraction 6. Encapsulation.

Platform independent :- Java code can be run on multiple platforms for eg. Windows, Linux, Sun Solaris. Java code is compiled by the compiler and converted to bytecode This bytecode is independent of platform thats why Java is Write Once Run Anywhere.

Secured :- Java is secure because no explicit pointers. and java program runs inside virtual machine sandbox.

Robust :- It uses strong memory management automatic garbage collection to get rid of unused objects
exception handling and type checking mechanism also.

Architecture-neutral : Java is archictecture neutral because there are no implementation dependent features.

Portable - Because it uses bytecode which runs on any platform.

High performance :- Java is faster than traditional interpreted languages.

Distributed :- Java is distributed because it facilitates users to create distributed applications

Multi-Threaded :- a thread is executed simultaneously in multithreaded environment. It doesn't occupy memory like processes

Dynamic :- Due to loading of classes dynamically. and automatic garbage collection

Q.8 What is JVM?

JVM (Java Virtual Machine) is a abstract machine. It is called as abstract machine because it is not physically present. It can run the programs which are converted to java bytecode

JVM performs below tasks:
- Loads code
- Verifies code
- Executes code
- Provides Runtime Environment

Q4 What is OOPS? Why it is called object oriented?

OOPS (Object-Oriented Programming System) is a methodology or paradigm to design program using classes and object.
It simplifies software developement and maintainence by providing some concepts

- classes
- objects
- Inheritence
- Polymorphism
- abstraction
- encapsulation

It is called as object oriented because as the name suggests it refers to the language that uses objects. The main aim of OOP is to bind together data and functions that operate on specific parts and no other code can access this data except the functions that are part of the object.

Q5 What is multiple inheritence? Explain in real time and programmatic way.

Multiple Inheritence :-
Multiple inheritence is type of inheritence in which a class can inherit properties of more than one parent class.

```
class A {
}

class B {
}

class C extends A, B {
    psvm (String args[]) { }
}
```

But this is not allowed in java.

So you can use multiple inherilence bay implementing Interfaces in java.

```
interface A {
        void show();
      }

interface B {
        void show();
      }

class C implements A, B
    {
        public void show() {
            A.super.show();
            B.super.show();
        }
        p s v m (String args[]) {
         C c = new C();
         c.show(); // valid.
      }
```

for example:

A class vehicle a interface Vehicle have method run() and 2nd interface Fly have method fly() we can implement both methods for a class Aeroplane which needs implementation of both methods.

16 Difference between abstract class and interface.

abstract class	interface
If we do know about partial implementation but partially then we should go for abstract class.	If we don't know about implementation just we have requirement specification then we should go for interface.
Every method present in abstract class need not be public abstract	Every method present interface is always public and abstract whether we are declaring or not.
every abstract class variable need not be public static final	every interface variable is public static final declared or not.
There are no restrictions on abstract class method modifiers.	we can't declare interface methods with the modifiers private, protected, final, static, synchronized, native, strictfp.
(need to initialize at the time of declaration or gives compilation error)	(not required to perform initialization with declaration)
Inside we can't take static and instance block	Inside abstract class we can have instance and static blocks.
Inside interface you can not have constructor	Inside we can have constructor in abstract class

Page No.
Date

Q.7 What is encapsulation ? Explain with real time example

Encapsulation in java is wrapping code and data together into a single unit.

For example : a mobile phone or a portable device.

we can use fully encapsulated class class in java by making all data members private Now we can use setter getter to set & get data in it.

The Java Bean class is example of fully encapsulated class.

Advantages : you can make class read only / write only

It provides you control over data

It is a way of achieving data hiding

It is better for unit testing.

What is polymorphism? and Inherilence?

The word polymorphism means having many forms In simple words we can say ability to perform ~~displa~~ operations in more than one form.

Real Life example : A mobile does a job of calling, playing music, video and gaming as per requirents so it can take different implementation for different tasks.

Mainly divided into two types 1. compile time polymorphism
2. Runtime Polymorphism

also known as earlybinding / static polymorphism

late binding / dynamic polymorphism.

TYP

Inherilence

Inherilence basically means one object acquires properties of another object by forming a relationship

IS-A relationship also known as Inherilence

By using extends keyword we can implement IS-A Relationship

The main advantage of IS-A Relationship is Reusability of the code.

Q 9 What is Method Overriding and Method Overloading?

	Overloading	Overridding
Method names	must be same	must be same
Arguments	must be different (at least order)	must be same (including order)
Return Type	No restrictions	must be same until 1.4 version but from 1.5 version onwards co-variant return types allowed
Access modifiers	No restrictions	Weakening is not allowed.
private, static final methods	can be overloaded	cannot be overidden.
Method Resolution	always taken care at compile time based on refrence type.	always take care by JVM based on runtime object
Also called as	Compile time polymorphism Early binding.	Runtime Polymorphism Late binding