**Q8**

**class** is a user defined blueprint or prototype from which **objects** are created also contained methods.

**object** is an element (or instance) of a **class**; **objects** have the behaviors of their **class**.

Q9

constructor is a block of codes similar to the method. It is called when an instance of the object is created.

\*Default Constructor

\*No-Argument Constructor

\*Parameterized Constructor

Q11

Used to implement an interface

Q12

Final can be used: data, methods,class

Q13

inheritance , polymorphism, encapsulation, abstraction

Q14

static block will be executed

Q15

No

Q16

interface

 interface are implicitly abstract and cannot have implementations.

All methods of an Interface are abstract

**Abstract**

In the case of an abstract class, a class may extend only one abstract class

An abstract class can have non-abstract methods

An abstract class can have any visibility: public, private, protected

Q17

void keyword specifies that the function does not return a value

used at main() method

Q18

Compile time polymorphism ,same name of mothed defiant parameter

Return same or deferent type

Q19

you can declare a method with the same signature in a subclass

Q20

1. Default Private
2. Protected
3. Public

Q21

local variable

declared within the body of a method

instance variable

used by Objects to store their states

Q22

**Method** name should not be of the same name as that

**Constructor** should be of the same name as that of class

**Constructors** are invoked implicitly wherea

**methods** are invoked explicitly

**super()** - refers immediate parent class instance

this**()** - refers current class instance. Can be used to invoke immediate parent class method

Q23

Mothed overloading

Class Animal{

Public void add(){

}

Public void add(int x ,int y){

}

Public void add( int x,int y, int z){

}

**overloading** is being done in the same class while

compile polymorphism

Mothed overriding

Class Animal{

Public vooid add() {

}

Class Dog extands Anumal{

Public vooid add() {

}

**overriding** base and child classes are required

runtime polymorphism