Assignment -2

① **Overloding**

i, Binding of overloaded method call to its definition happens at compile time

ii, Stack, final method can be overloaded

Ex:-
```
Class Dog{
    Public void bark(){
        system.out.point/n"
              woof"}.}

    //overloding method.
    Public void bark (intnum){
        for(inti=0; i<num; i++)
            system.o.p ("woof");

    }
}
```

Overriding

i, Binding of overriden method call to its definition happen at run time

(iii) It is declared a some static method in child the same methodo class in static inp waate final child cannot overrided the privede final methods of their bases class.

eg:-
```
Class Dog{
  Public void bark(){
    sys tem.o.p ("woof");}
  }
class Hound extends dog}
  Public void snf(){
    s.o.p/n ("sinff")}}
  Public void bark()2 {
    s.o.p/n ("bowl")!}}
class test{
  Public static void main
         s tein [args){
    Dog .dog = new Hound();
      dog. back(); }
  }
```

Scanned with CamScanner

```
2) ClassA {
    Public void show() {
        System.out. point/n ("ind A) ; }
    }
     class B {
      Public void Show() {
          system.out. point/n ("int B"); }

      }
       class c {
        Public void Show() {
             System.out . Point/n("int c"); }
           }
      Public void Config() {
           system.out point /n ("config") ; }

       Public void config() {
             System.out .point ln ("config"); }

       Public class Demo {
          Public static void main (String []args){
            A.obj 1= new B() ;
             obj 1 = show();
              A.boj 2 = new1 ();
                obj2 = config() ;
                obj 1=new1();
               ① obj1.show(); }
                }
```

①:- Because obj we linking in runtime, this will work on show only on runtime. so obj, gives on that output

runtime polymorphism.

③ Abstract methods and Classes :-

An abstract class is a class that is declared abstract-it may not include abstract method Abstract class cannot be instantiated, but they can be subclassed.

→ An abstract method is a method that is declared without an implementation without braces and the followed by a semicolon like this :-

abstract void move double deltar, double deltax)

if a class includes abstract methods, then the class itself must be declared abstract, as in

```
Public abstract class Graphic object {
        // declare fields
        // declare non abstract methods
    abstract void draw ();
}
```

→ An abstract class is subclassed, the subclass usally provides Implementation for all of the abstract methods in its parent class However it does not, then the the subclass must also be declared atleast.

(4) You can intialize a final variable when it is declared. this approach is the most common. A final variable is called blank final variables, if it is not intialized while declaration, Below are the two ways. to intitize a blank final variable.

(*) A blank final variable can be initialized constructor. If you have more than one constructor if your class then it more than it most be intialized in all of them, other wise compiler time will be throwon.

(*) Ablank final static variable can be intialized inside static block.

Scanned with CamScanner