Q1. Explain super keyword in java programming?
Ans:
Using super we can call parent class constructor, we can access parent class member data, we can call parent class member function inside child class

1. Call parent class constructor
Syntax:
super(parameter1,parrameter2);

2. Access Parent class member data
Syntax:
super.memberDataName;

3. Call parent class method
Syntax:
super.methodName();

Note: super must be first line of the the block

```
class Point{
public int x;
public int y;//instance variable
public int z=101;
public Point(){
System.out.println("Point class Default
Constructor is called");
}

public Point(int x,int y){
this.x=x;
this.y=y;
System.out.println("Point class Parameterized
Constructor is called");
}

void showData(){
System.out.println("X_CO : "+x);
System.out.println("Y_CO : "+y);

}
void hi(){
System.out.println("Hi... Method is Called");

int x=111;
int y=222;
System.out.printf("\nx=%d  Y=%d ",this.x,this.y);


}
```

```
}
class Circle extends Point{
float r;
int z=102;
public Circle(){
System.out.println("Circle class Default
Constructor");
}
public Circle(int x,int y,float r){
super(x,y);//call  parent class constructor
this.r=r;
System.out.println("Circle class Parameterized
Constructor");
}
void showData(){
super.hi();
System.out.println("X_CO : "+x);
System.out.println("Y_CO : "+y);
System.out.println("Radius is : "+r);
System.out.println("circle z= : "+z);
System.out.println("Point class z= : "+super.z);
}
```

```
public static void main(String args[]){
//Circle c1=new Circle();
//Point class default constructor (1)and Circle
class default constructor(2)
Circle c2=new Circle(11,22,5.6f);
c2.showData();
}


}
```

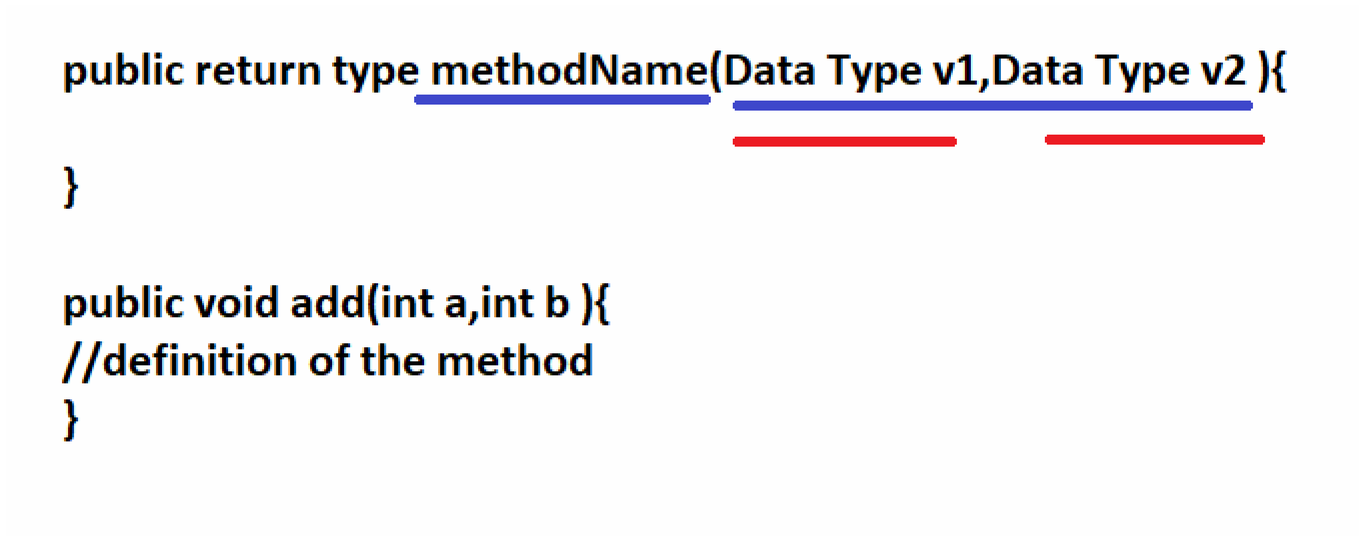
Q2. Explain Polymorphism in java programming?
Ans:
One Name Multiple form is known as Polymorphism
There are two types of Polymorphism in java
1. Compile Time Polymorphism:(Method Overloading): Decide at compile time which method is called(Static Binding/Early Binding)

2. Run time Polymorphsm(Dynamic Binding/Late Binding) : (Method Overriding):
Decide at run time which method is called

Q3. Explain Method Overloading in java Programming?

Ans: Method overloading means defined multiple methods with same name but different signature.

Points Regarding Method Overloading



```
public return type methodName(Data Type v1,Data Type v2 ){

}

public void add(int a,int b ){
//definition of the method
}
```

1. Same Method Name: All overloaded methods must have same name
2. Different Parameters: Overloaded methods differ in

A. Number of Parameter

B. Data types of Parameters

C. Order of Parameters

3. Return type: Return type can be different or same but it does not affect the concept of method overloading

```
class Test{

public void add(){
int a,b,c;
a=1;
b=2;
c=a+b;
System.out.println("Addition without argument : "+c);
}
public void add(int a,int b){
int c;
c=a+b;
System.out.println("Addition with two int argument : "+c);

}
public void add(int a,float b){
float c;
c=a+b;
System.out.println("Addition with two int,float argument :
"+c);

}
public void add(float a,int b){
float c;
c=a+b;
System.out.println("Addition with two float,int argument :
"+c);
}
public void add(String a,String b){
int c;
c=Integer.parseInt(a)+Integer.parseInt(b);
System.out.println("Addition with two String argument :
"+c);

}

public static void main(String args[]){
Test t=new Test();
t.add();
t.add(10,20);
t.add(5,2.5f);
t.add(15.5f,10);
t.add("1","1");

}


}
```