```
class Emp
{
void add(int a,int b)
{ System.out.println("Int : " + (a+b)); }

void add(float a,float b)
{ System.out.println("Float : " + (a+b)); }

void add(char a,char b)
{ System.out.println("Character : " + (a+b)); }

void add(String a,String b)
{ System.out.println("String  : " + (a+b)); }
}

class Overload1
{
public static void main(String args[])
{
Emp e=new Emp();
e.add(10,20);
e.add(100.50F,20.50F);
e.add('A','B');
e.add("Welcome","Hello");
}
}
----------------------------------------------------------------------------------------------------------------------
import java.io.*;

class calculation
{
public void add(int a,int b)
{
int c=a+b;
System.out.println("Add of 2 Int is : "+c);
}

public void add(float a,float b)
{
float c=a+b;
System.out.println("Add of 2 Float is : "+c);
}

public void add(String a,String b)
```

```
{
String c=a+b;
System.out.println("Add of 2 String is : "+c);
}
}

class overload2
{
public static void main(String args[])
{
calculation c=new calculation();
c.add(10,20);
c.add(40.0F,60.0F);
c.add("Have a ","Nice Day");
}
}
-------------------------------------------------------------------------------------------------------------------
import java.io.*;

class super1
{
int x;
super1(int x)
{
this.x=x;
}
void display()
{
System.out.println("1.Super x="+x);
}
}
class sub1 extends super1
{
int y;
sub1(int x,int y)
{
super(x);
this.y=y;
}
void display()
{
System.out.println("2.Super x = "+x);
System.out.println("Sub y = "+y);
}
```

```
}
class override1
{
public static void main(String args[])
{
sub1 s1=new sub1(100,200);
s1.display();
}
}
/*It will override the method of super class in sub class. When we want an object to respond to  the
same method but have different behaviour when that method is called.That means,we should override
the method defined in the super class. This is possible by defining a method in the sub class that has
same name,same arguments and same return type as a method in the super class.Then, when that
method is called , the method defined in the sub class is invoked and executed instead of the one in the
super class.This is known as overriding.*/
```

```
class toString1
{
public String toString()
{
String s="This is Overridden";
return s;
}

public static void main(String arg[])
{
toString1 b=new toString1();
System.out.println(b);
}
}
```
