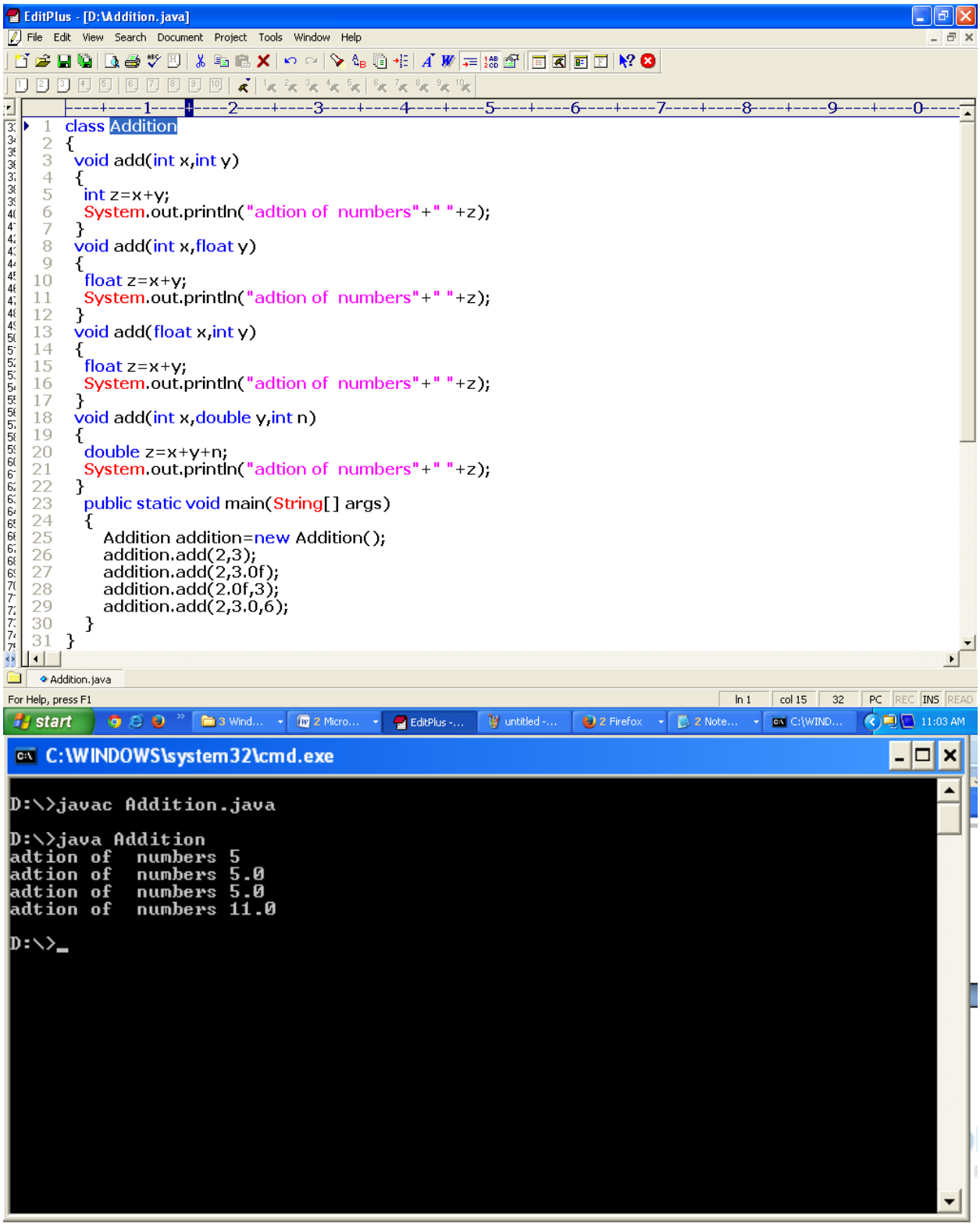
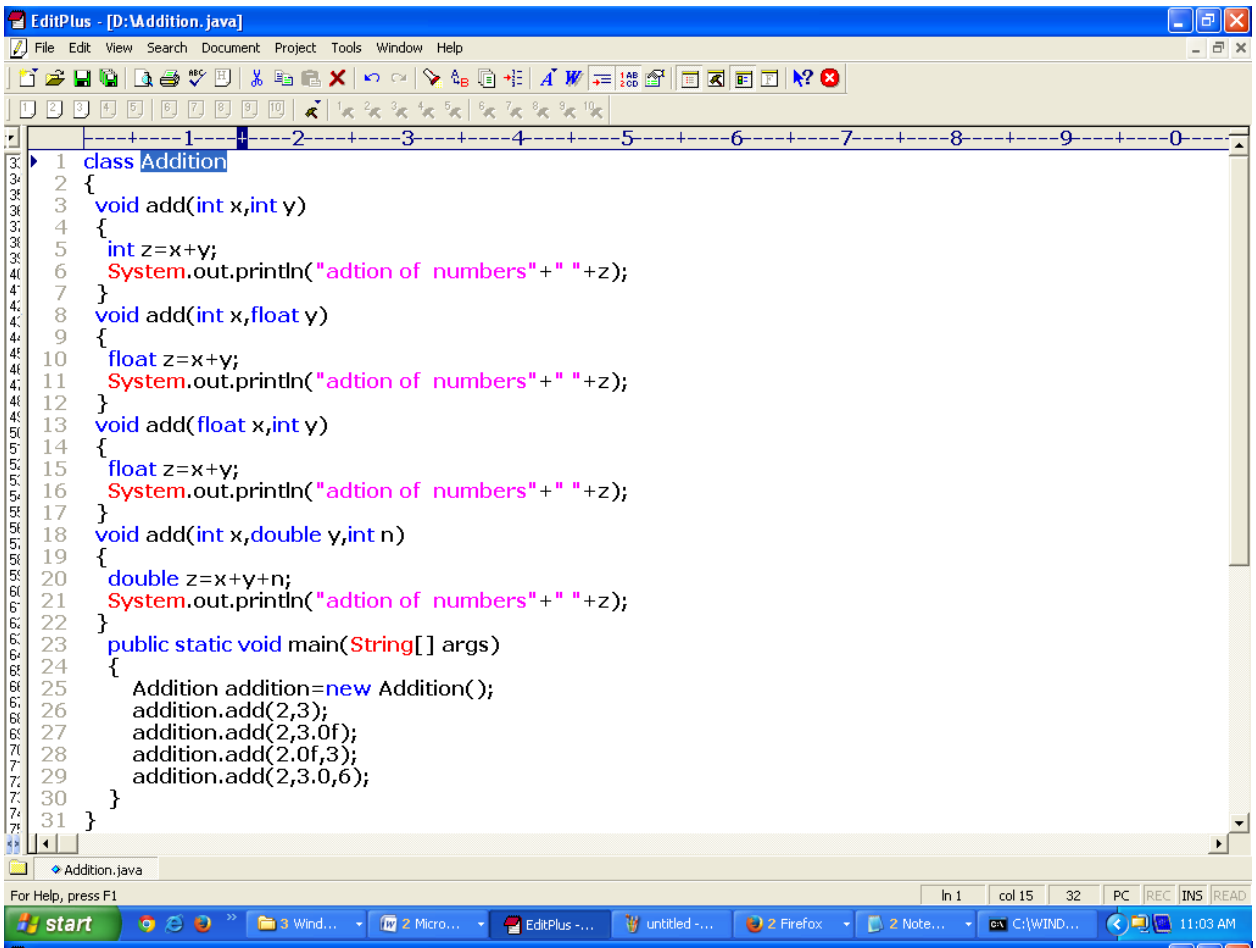
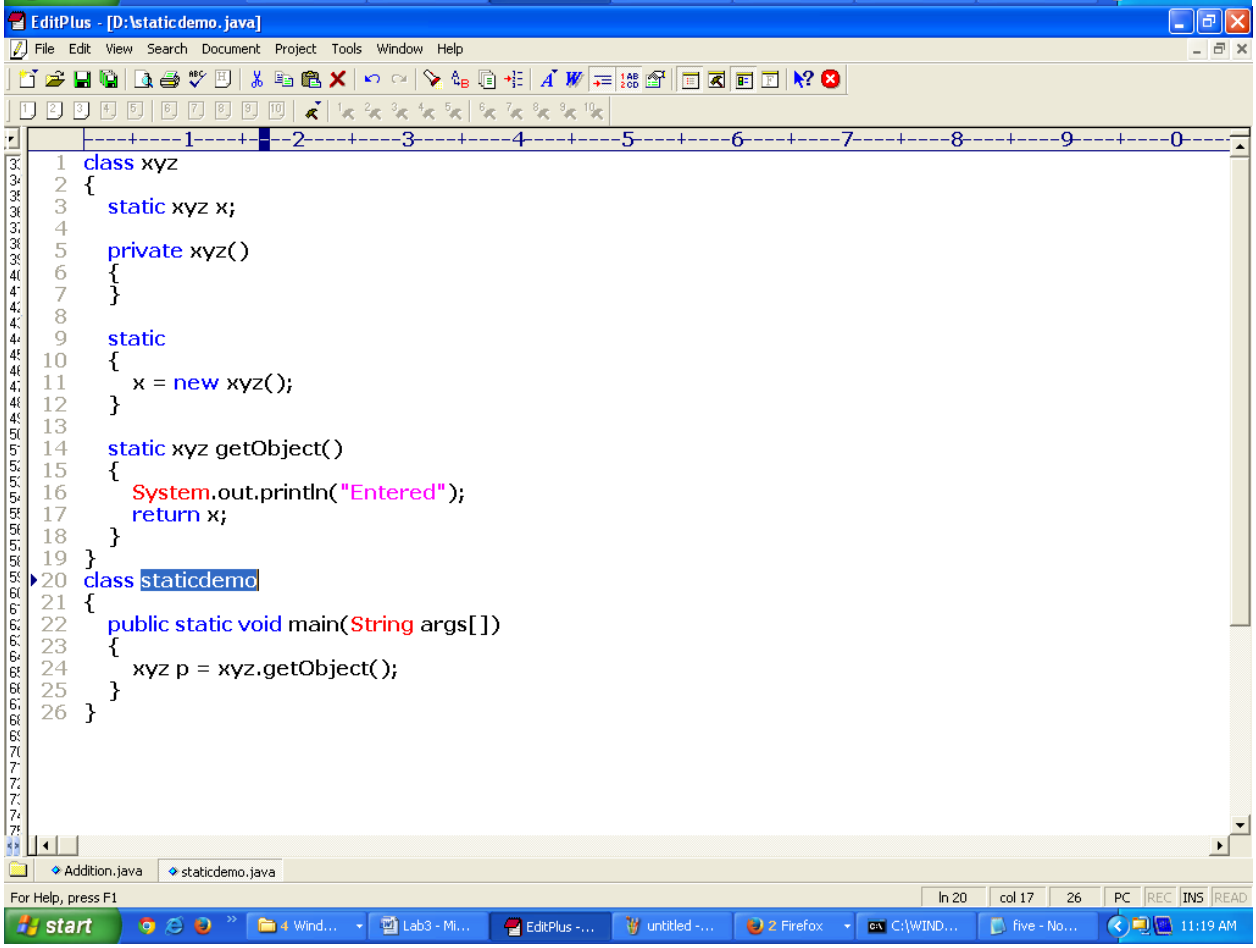
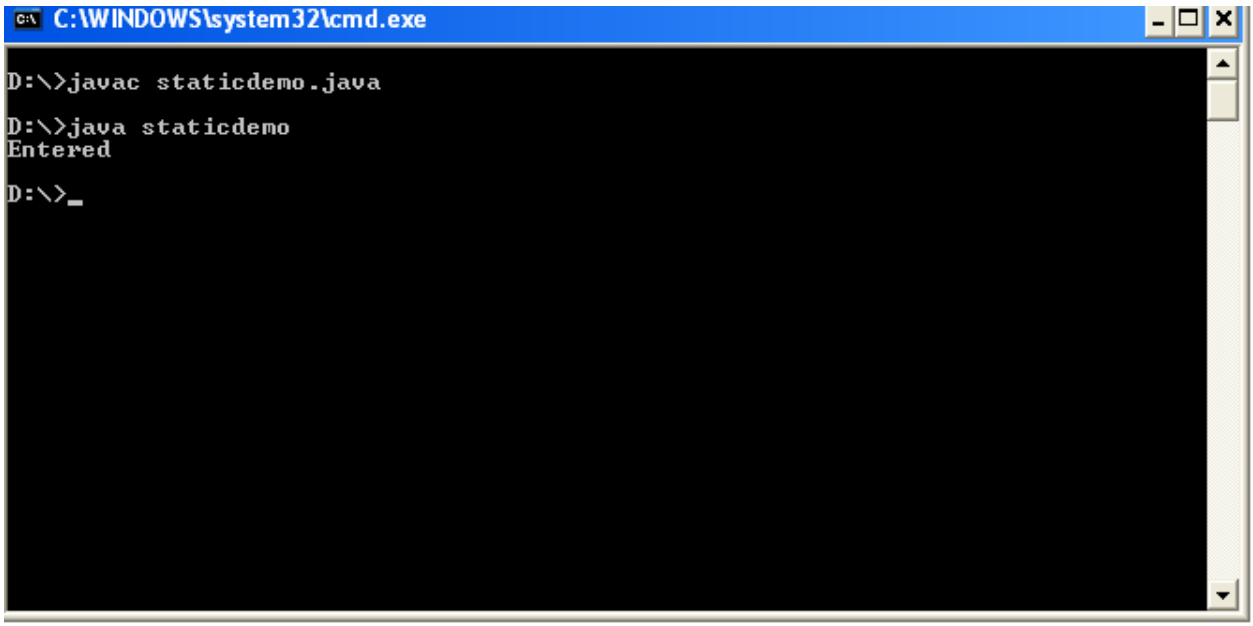
` Lab-3 Assignments

Introducing Constructor,OverLoading Constructors & Methods




EditPlus - [D:\Addition.java]
File Edit View Search Document Project Tools Window Help
class Addition
{
void add(int x,int y)
{
int z=x+y;
System.out.println("adtion of numbers"+" "+z);
}
void add(int x,float y)
{
float z=x+y;
System.out.println("adtion of numbers"+" "+z);
}
void add(float x,int y)
{
float z=x+y;
System.out.println("adtion of numbers"+" "+z);
}
void add(int x,double y,int n)
{
double z=x+y+n;
System.out.println("adtion of numbers"+" "+z);
}
public static void main(String[] args)
{
Addition addition=new Addition();
addition.add(2,3);
addition.add(2,3.0f);
addition.add(2.0f,3);
addition.add(2,3.0,6);
}
}
Addition.java
For Help, press F1
ln 1 col 15 32 PC REC INS READ
start
11:03 AM


EditPlus - [D:\staticdemo.java]
File Edit View Search Document Project Tools Window Help
class xyz
{
static xyz x;
private xyz()
{
}
static
{
x = new xyz();
}
static xyz getObject()
{
System.out.println("Entered");
return x;
}
}
class staticdemo
{
public static void main(String args[])
{
xyz p = xyz.getObject();
}
}
Addition.java
staticdemo.java
For Help, press F1
ln 20 col 17 26 PC REC INS READ
start
11:19 AM



Assignments To Solve

1. Write an Account class with default constructor, parameterised constructor and methods toString(), deposit(int amt), withdraw(int amt). withdraw method should take care of insufficient balance. Accept the values from the user..Display the details of various Accounts
2. Write a class Addition2 with add method (overloading) using varargs and enhanced for loop
3. Declare three array of integer types ,Accept values for two array.Perform arithmetic operation on both array values and store corresponding values in third array.
   E,,g, : c[0]=a[0]+b[0]
4. Write a class stack. Use methods push and pop to store and retrieve elements from stack(hint: create an array of int to store the elements)
5. Write a singleton class(Singleton class is is a class which has only one object)