**List of JAVA Program**

**Program 8**

// This class defines an integer stack that can hold 10 values.

class Stack

{

int stck[] = new int[10];

int tos;

// Initialize top-of-stack

Stack()

{

tos = -1;

}

// Push an item onto the stack

void push(int item)

{

if(tos==9)

System.out.println("Stack is full.");

else

stck[++tos] = item;

}

// Pop an item from the stack

int pop()

{

if(tos < 0)

{

System.out.println("Stack underflow.");

return 0;

}

else

return stck[tos--];

}

}

class TestStack

{

public static void main(String args[])

{

Stack mystack1 = new Stack();

Stack mystack2 = new Stack();

// push some numbers onto the stack

for(int i=0; i<10; i++)

mystack1.push(i);

for(int i=10; i<20; i++)

mystack2.push(i);

// pop those numbers off the stack

System.out.println("Stack in mystack1:");

for(int i=0; i<10; i++)

System.out.println(mystack1.pop());

System.out.println("Stack in mystack2:");

for(int i=0; i<10; i++)

System.out.println(mystack2.pop());

}

}

**Program 9**

// Demonstrate method overloading.

class OverloadDemo

{

void test()

{

System.out.println("No parameters");

}

// Overload test for one integer parameter.

void test(int a)

{

System.out.println("a: " + a);

}

// Overload test for two integer parameters.

void test(int a, int b)

{

System.out.println("a and b: " + a + " " + b);

}

// overload test for a double parameter

double test(double a)

{

System.out.println("double a: " + a);

return a\*a;

}

}

class Overload

{

public static void main(String args[])

{

OverloadDemo ob = new OverloadDemo();

double result;

// call all versions of test()

ob.test();

ob.test(10);

ob.test(10, 20);

result = ob.test(123.25);

System.out.println("Result of ob.test(123.25): " + result);

}

}

This program generates the following output:

No parameters

a: 10

a and b: 10 20

double a: 123.25

Result of ob.test(123.25): 15190.5625

**Program 10**

/\* Here, Box defines three constructors to initialize

the dimensions of a box various ways.

\*/

class Box

{

double width;

double height;

double depth;

// constructor used when all dimensions specified

Box(double w, double h, double d)

{

width = w;

height = h;

depth = d;

}

// constructor used when no dimensions specified

Box()

{

width = -1; // use -1 to indicate

height = -1; // an uninitialized

depth = -1; // box

}

// constructor used when cube is created

Box(double len)

{

width = height = depth = len;

}

// compute and return volume

double volume()

{

return width \* height \* depth;

}

}

class OverloadCons

{

public static void main(String args[])

{

// create boxes using the various constructors

Box mybox1 = new Box(10, 20, 15);

Box mybox2 = new Box();

Box mycube = new Box(7);

double vol;

// get volume of first box

vol = mybox1.volume();

System.out.println("Volume of mybox1 is " + vol);

// get volume of second box

vol = mybox2.volume();

System.out.println("Volume of mybox2 is " + vol);

// get volume of cube

vol = mycube.volume();

System.out.println("Volume of mycube is " + vol);

}

}

The output produced by this program is shown here:

Volume of mybox1 is 3000.0

Volume of mybox2 is -1.0

Volume of mycube is 343.0