# Lab # 3
# CP2530
# Stack Programming

Main.java

```java
//this creates class pair to enable the stack to have a pair of integers to push and pop
class Pair {
    int first;
    int second;

    public Pair(int first, int second) {
        this.first = first;
        this.second = second;
    }

    public String toString() {
        return "(" + first + "," + second + ")";
    }
}
//class that creates the stack
class Stack {
    static final int MAX = 1000;
    int top;
    Pair[] array = new Pair[MAX];

    public Stack() {
    top = -1;
    }

    public boolean isEmpty() {
    return (top < 0);
    }
    //push method that adds integers to the stack
    public boolean push(int x, int y) {
        if (top >= (MAX - 1)) {
            System.out.println("Stack overflow");
            return false;
        } else {
            Pair pair = new Pair(x, y);
            array[++top] = pair;
```

Output

```
java -cp /tmp/o5Jft71Jfm Main
(1,34) pushed into stackStack: (1,34)
(2,5) pushed into stack
Stack: (2,5) (1,34)
(50,34) pushed into stack
Stack: (50,34) (2,5) (1,34)
(101,17) pushed into stack
Stack: (101,17) (50,34) (2,5) (1,34)
(14,7) pushed into stack
Stack: (14,7) (101,17) (50,34) (2,5) (1,34)
(14,7) popped from stack
Stack: (101,17) (50,34) (2,5) (1,34)
Popped item: (14,7)
(101,17) popped from stack
Stack: (50,34) (2,5) (1,34)
Popped item: (101,17)
(50,34) popped from stack
Stack: (2,5) (1,34)
Popped item: (50,34)
(2,5) popped from stack
Stack: (1,34)
Popped item: (2,5)
(1,34) popped from stack
Stack:
Popped item: (1,34)
```

```
            array[++top] = pair;
            System.out.println(pair + " pushed into stack");
            displayStack();
            return true;
        }
    }
    //pop method that removes integers from the stack
    public Pair pop() {
        if (top < 0) {
            System.out.println("Stack underflow");
            return null;
        } else {
            Pair pair = array[top--];
            System.out.println(pair + " popped from stack");
            displayStack();
            return pair;
        }
    }
    //method to display the current stack and its elements
    public void displayStack() {
        System.out.print("Stack: ");
        for (int i = top; i >= 0; i--) {
        System.out.print(array[i] + " ");
        }
        System.out.println();
    }
}
//main method that pushes the integer pairs through the stack and then pops them
class Main {
    public static void main(String[] args) {
        Stack stack = new Stack();
        stack.push(1, 34);
        stack.push(2, 5);
        stack.push(50, 34);
        stack.push(101, 17);
```

```
        stack.push(101, 17);
        stack.push(14, 7);
        System.out.println("Popped item: " + stack.pop());
        System.out.println("Popped item: " + stack.pop());
        System.out.println("Popped item: " + stack.pop());
        System.out.println("Popped item: " + stack.pop());
        System.out.println("Popped item: " + stack.pop());
    }
}
```