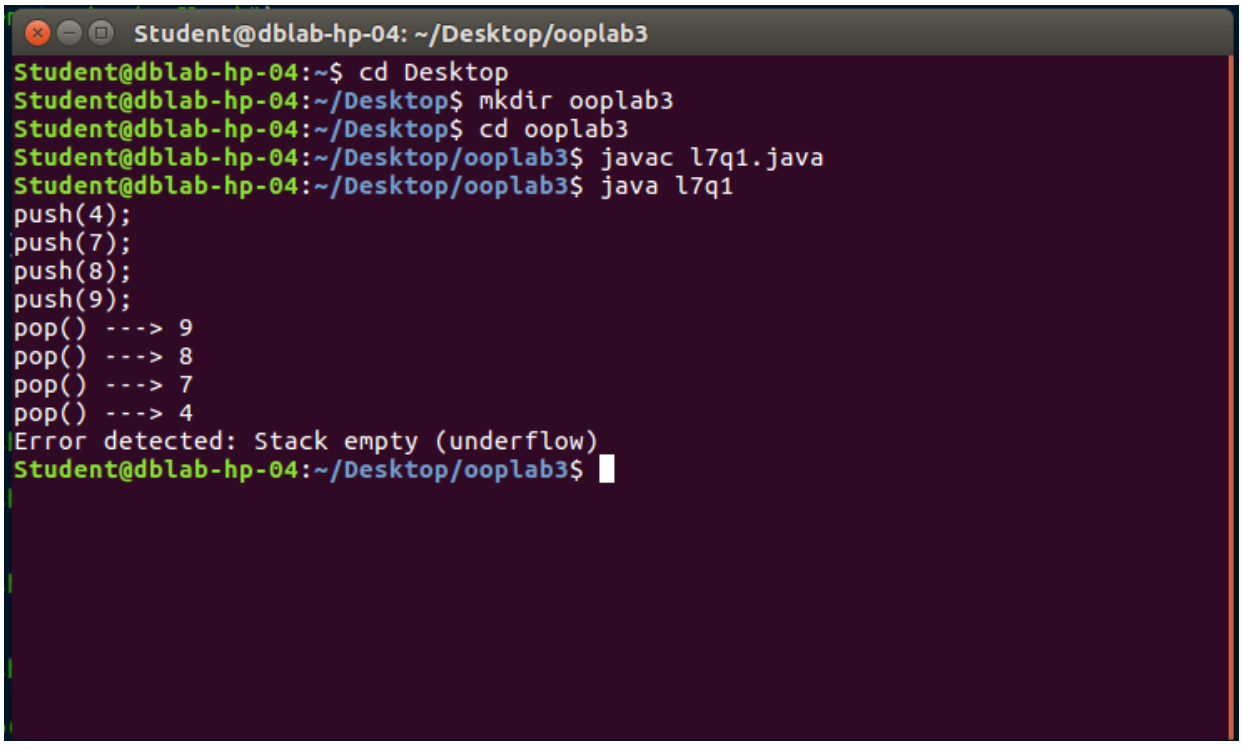
Ayush Goyal
190905522

OOP Lab 7 (Session 3)

Q1)

```
class pushException extends Exception{
	public pushException( String message ){
		super(message);
	}
}

class popException extends Exception{
	public popException( String message ){
		super(message);
	}
}

interface Stack{
	public void push( int x ) throws pushException;
	public int  pop ( ) throws popException;
}

class ArrayStack implements Stack{
	public int[] item;
	public int stackTop;
	public ArrayStack( int size ){
	item = new int[size];
	stackTop = 0;
	}

	public void push( int x ) throws pushException{
		if ( stackTop == item.length ){
			throw new pushException("Stack overflow");
		}
		item[stackTop] = x;
		stackTop++;
	}

	public int  pop ( ) throws popException{
		int returnItem;
		if ( stackTop == 0 ){
			throw new popException("Stack empty (underflow)");
		}
		returnItem = item[ stackTop-1 ];
		stackTop--;
		return returnItem;
	}
}

public class l7q1{
	public static void main( String[] args ){
		int   x;
```

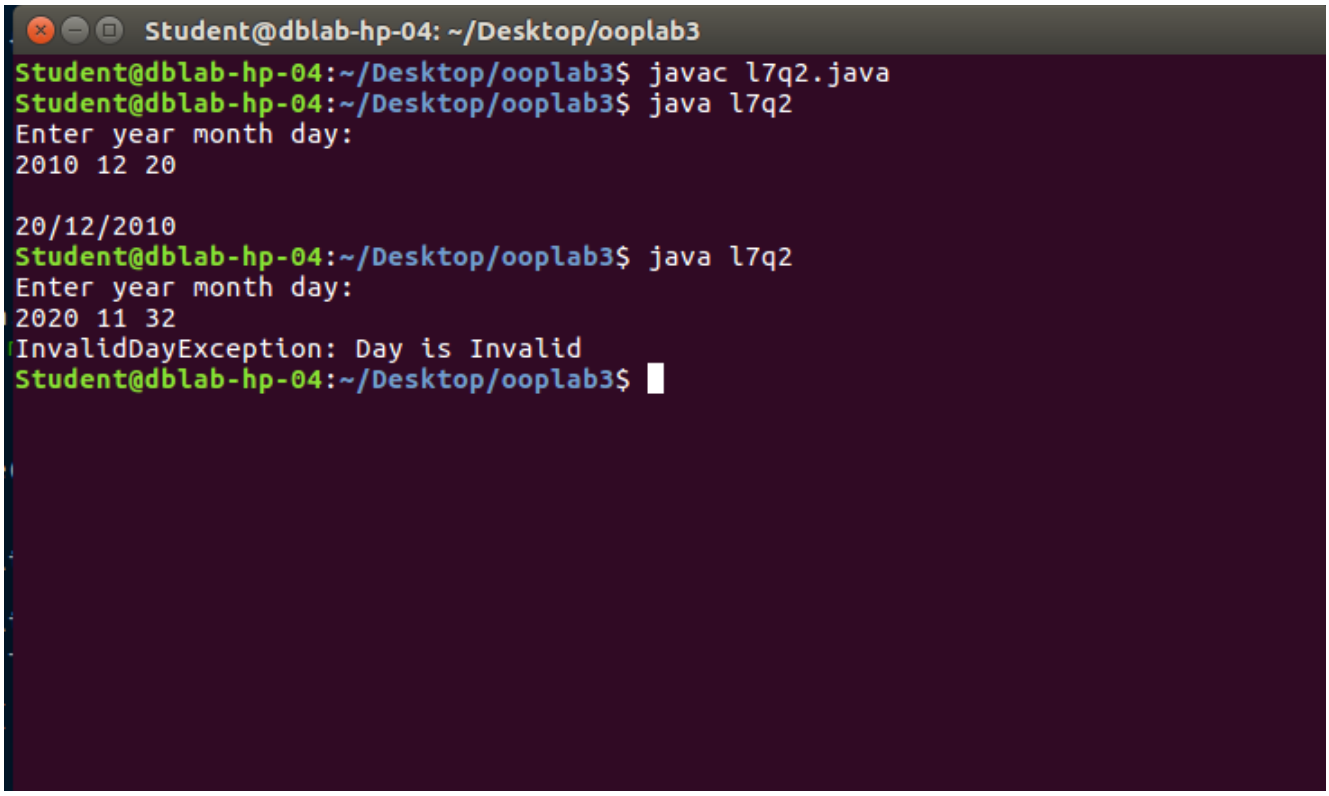
```
                Stack s;
                s = new ArrayStack( 6 );
                try{
                        x = 4;
                        s.push(x);
                        System.out.println("push(" + x + ");");
                    x = 7; s.push(x);
                        System.out.println("push(" + x + ");");
                    x = 8;
                        s.push(x);
                        System.out.println("push(" + x + ");");
                    x = 9;
                        s.push(x);
                        System.out.println("push(" + x + ");");
                    x = s.pop();
                        System.out.println("pop() ---> " + x );
                    x = s.pop();
                        System.out.println("pop() ---> " + x );
                    x = s.pop();
                        System.out.println("pop() ---> " + x );
                    x = s.pop();
                        System.out.println("pop() ---> " + x );
                    x = s.pop();
                        System.out.println("pop() ---> " + x );
                } catch ( pushException e ) {
                        System.out.println("Error detected: " + e.getMessage() );
                        System.exit(1);
                        }
                  catch ( popException e )  {
                      System.out.println("Error detected: " + e.getMessage() );
                        System.exit(1);
                        }
        }
}
```



Q2)

```
import java.util.*;
class CurrentDate{
	int day,month,year;
	CurrentDate() {
		System.out.println("\nNo Date Initialised.");
	}
	CurrentDate(int d,int m,int y) {
		day=d;
		month=m;
		year=y;
	}
	boolean checkmonth() {
		if(month<1 || month >12)
			return false;
		else
			return true;
	}
	boolean checkday() {
		boolean state;
		if(day<1 || day>31 )
			state=false;
		else{
			switch (month){
				case 1:
				case 3:
				case 5:
				case 7:
				case 8:
				case 10:
				case 12:
					if(day > 31){
						state= false;
					}
					else
						state= true;
					break;
				case 4:
				case 6:
				case 9:
				case 11:
					if(day > 30){
						state= false;
					}
					else
						state= true;
					break;
				case 2:
					if(year % 4 != 0 && day > 28){
						state= false;
					}
```

```
                                        else
                                                return true;
                                        if(day > 29){
                                                state= false;
                                        }
                                default:        state= false;
                        }
                }
                return state;
        }
}

class InvalidDayException extends Exception{
        public InvalidDayException(String s) {
                super(s);
        }
}

class InvalidMonthException extends Exception{
        public InvalidMonthException(String s) {
                super(s);
        }
}


class l7q2{
        static void createDate(){
                int d,m,y;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter year month day:");
                y = sc.nextInt();
                m = sc.nextInt();
                d = sc.nextInt();
                CurrentDate a = new CurrentDate(d,m,y);
                try{
                        if(! a.checkmonth())
                                throw new InvalidMonthException("Month is Invalid");
                        if(! a.checkday())
                                throw new InvalidDayException("Day is Invalid");
                        System.out.println("\n"+a.day+"/"+a.month+"/"+a.year);
                }
                catch(InvalidMonthException e){
                        System.out.println(e);
                }
                catch(InvalidDayException e){
                        System.out.println(e);
                }
        }
        public static void main(String[] args){
                createDate();
        }
}
```



Q3)

```
import java.util.Scanner;
import java.text.*;
import java.util.*;
class seatsFilledException extends Exception{
   public seatsFilledException( String message )   {
      super(message);
    }
}
class StudentRecord {
    static int count=0;
    int regNo;
    String fullName;
    GregorianCalendar dateJoined;
    int sem;
    float gpa;
    float cgpa;
    StudentRecord () {
        count=count+1;
        regNo = 0;
        fullName = "";
        dateJoined = new GregorianCalendar();
        sem = 0;
        gpa = 0.f;
        cgpa = 0.f;
  }
    StudentRecord (String fullName, int y, int m, int d, int sem, float gpa, float cgpa) {
        if(count/10 == 0)
            regNo = (y%100)*10+count;
        else
```

```
                regNo = (y%100)*100+count;
        this.fullName = fullName;
       this.dateJoined = new GregorianCalendar(y, m-1, d);
       this.sem = sem;
       this.gpa = gpa;
       this.cgpa = cgpa;
       count=count+1;
  }
   public void display () {
       SimpleDateFormat dateFormat = new SimpleDateFormat("MM/dd/yyyy");
        System.out.println("Name :" + fullName + "\t     Reg No :" + regNo + "\tDate Joined :" +
dateFormat.format(dateJoined.getTime()) + "\t   Semester :" + sem + "\t       GPA :" + gpa + "\t
CGPA :" + cgpa);
       System.out.println();
   }
   public void input () throws seatsFilledException{
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Name: ");
            fullName = sc.nextLine();
            System.out.print("Enter Date Joined (yyyy mm dd): ");
            int y,m,d;
            y=sc.nextInt();
            m=sc.nextInt()-1;
            d=sc.nextInt();
            dateJoined = new GregorianCalendar(y,m,d);
            System.out.print("Enter Semester: ");
            sem = sc.nextInt();
            System.out.print("Enter GPA: ");
            gpa = sc.nextFloat();
            System.out.print("Enter CGPA: ");
            cgpa = sc.nextFloat();
            System.out.println();
            if(count/10 == 0)
                    regNo = (y%100)*10+count;
              else if(count>25)
                        throw new seatsFilledException("Seats Filled");
              else
                        regNo = (y%100)*100+count;
    }
}


public class l7q3{
        public static void main (String args[]) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Creating 25 random records to generate error...");
                StudentRecord std[] = new StudentRecord[30];
                for(int i=0;i<25;i++){
                        std[i] = new StudentRecord("abc", 2012, 11, 23, 4, 9, 9);
                }
                try{
                        System.out.println("Enter 26th record to generate error:\n");
```

```
                    std[25] = new StudentRecord();
                    std[25].input();
                }
                catch(seatsFilledException e)  {
                    System.out.println("Error detected: " + e.getMessage() );
                    System.exit(1);
                }
        }
}
```