190905514 **MOHAMMAD TOFIK**

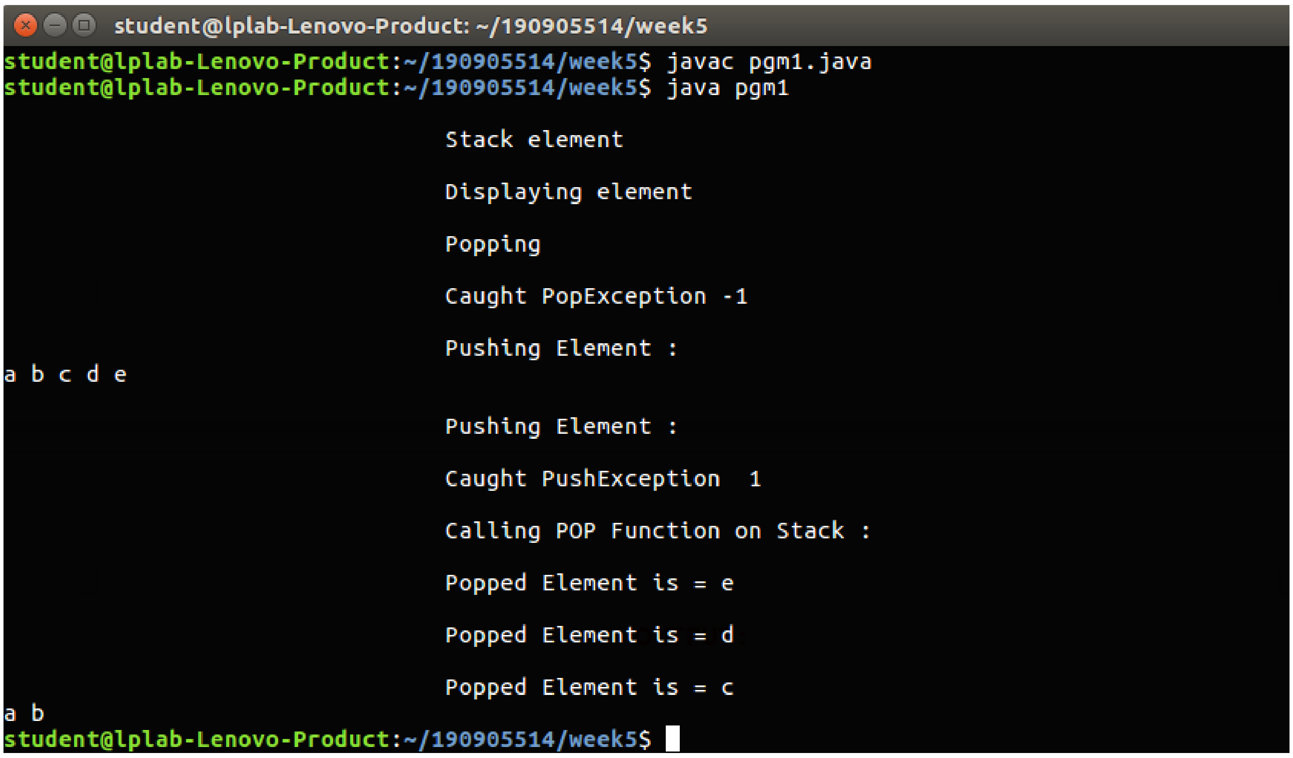
SESSION 3 LAB NO 7 :

1.Design a stack class. Provide your own stack exceptions namely Push Exception and Pop Exception, which throw exceptions when the stack is full and when the stack is empty respectively. Show the usage of these exceptions in handling a stack object in the main.

pgm1.java

```
class PushException extends
Exception
{
private int strange;
public PushException(int number)
{
this.strange = number; }
public int getStack()
{ return strange;
} }
class PopException extends Exception
{
private int strange;
public PopException(int number)
{
this.strange = number; }
public int getStack()
{ return strange;
} }
class Stack
{
private char item[];
private int top;
private int size;
public Stack()
{ this.item = new char[0];
this.top = -1; this.size =
0;
}
public Stack(int size){
this.size = size; this.item =
new char[size]; this.top = -1; }
public boolean isEmpty()
{
if(this.top == -1)
return (true);
return (false);
}
public boolean isFull()
{
if(this.top == this.size -1)
return (true);
return (false);
}
public boolean push(char element) throws PushException
{ if(this.isFull())
{ throw new PushException(1);
}
this.item[++this.top] = element;
return (true);
}
public char pop() throws PopException
{
if(this.isEmpty())
{ throw new PopException(-1);
}
return(this.item[this.top--]);
}
public void display()
{ if(this.isEmpty()) return; for(int i= 0; i < this.top + 1; i++)
System.out.print(String.format("%c ", this.item[i])); System.out.println("");
} }
class pgm1
{
public static void main(String[] args)
{

Stack myStack = new Stack(5);
System.out.println( "\n\t\t\t\tStack element");
System.out.println( "\n\t\t\t\tDisplaying element");
myStack.display();
System.out.println( "\n\t\t\t\tPopping");
try{ char element = myStack.pop();
System.out.println("\n\t\t\t\tPopped element is = " + element);
}catch(PopException e)
{
System.out.print("\n\t\t\t\tCaught PopException ");
System.out.println(e.getStack());
}
System.out.println( "\n\t\t\t\tPushing Element : "); try{
myStack.push('a');
 myStack.push('b');
 myStack.push('c');
 myStack.push('d');
 myStack.push('e');
myStack.display();
System.out.println("\n\t\t\t\tPushing Element : ");
myStack.push('f');}catch(PushException e)
{
System.out.print("\n\t\t\t\tCaught PushException  ");
System.out.println(e.getStack());
}
System.out.println("\n\t\t\t\tCalling POP Function on Stack : "); try{
System.out.println("\n\t\t\t\tPopped Element is = " + myStack.pop());
System.out.println("\n\t\t\t\tPopped Element is = " + myStack.pop());
System.out.println("\n\t\t\t\tPopped Element is = " + myStack.pop());
}catch(PopException e)
{
System.out.print("\n\t\t\t\tCaught PopException : ");
System.out.println(e.getStack());
}
 myStack.display();
}
}
```

OUTPUT :



```
student@lplab-Lenovo-Product: ~/190905514/week5
student@lplab-Lenovo-Product:~/190905514/week5$ javac pgm1.java
student@lplab-Lenovo-Product:~/190905514/week5$ java pgm1
                                Stack element

                                Displaying element

                                Popping

                                Caught PopException -1

                                Pushing Element :
a b c d e

                                Pushing Element :

                                Caught PushException  1

                                Calling POP Function on Stack :

                                Popped Element is = e

                                Popped Element is = d

                                Popped Element is = c
a b
student@lplab-Lenovo-Product:~/190905514/week5$
```

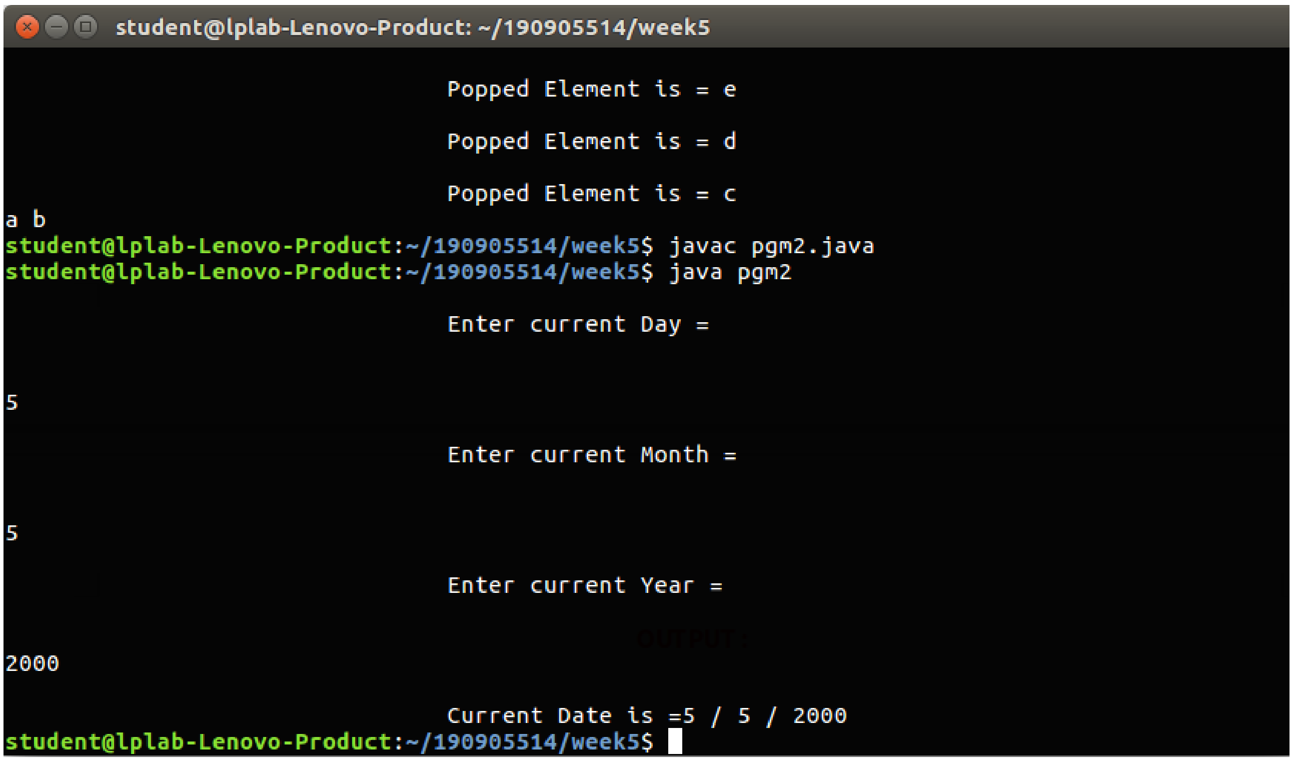
2.Define a class CurrentDate with data members day, month and year. Define a method createDate() to create date object by reading values from keyboard. Throw a user defined exception by name InvalidDayException if the day is invalid and InvalidMonthException if month is found invalid and display current date if the date is valid. Write a test program to illustrate the functionality.

pgm2.java

```java
import java.util.Scanner;

class currentDate {
  int date;
  int month;
  int year;

  currentDate(int a, int b, int c) {
    date = a;
    month = b;
    year = c;
  }

  void createDate(int date, int month, int year)
      throws pgm2.InvalidDayException, pgm2.InvalidMonthException {
    switch (month) {
      case 1:
      case 3:
      case 5:
      case 7:
      case 8:
      case 12:
        if (date >= 1 && date <= 31) {
          this.date = date;
```

```
                this.month = month;
                this.year = year;
            } else
                throw new pgm2.InvalidDayException();
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            if (date >= 1 && date <= 31) {
                this.date = date;
                this.month = month;

            } else throw new pgm2.InvalidDayException();
            break;
        case 2:
            if (year % 4 == 0) {
                if (year % 100 == 0) {
                    if (year % 400 == 0) {
                        if (date >= 1 && date <= 24) {
                            this.date = date;
                            this.month = month;
                        } else throw new pgm2.InvalidDayException();
                    } else {
                        if (date >= 1 && date <= 28) {
                            this.date = date;
                            this.month = month;
                        } else throw new pgm2.InvalidDayException();
                    }
                    if (date >= 1 && date <= 29) {
                        this.date = date;
                        this.month = month;
                    } else throw new pgm2.InvalidDayException();
                } else {
                    if (date >= 1 && date <= 28) {
                        this.date = date;
                        this.month = month;
                    } else throw new pgm2.InvalidDayException();
                }
                break;
            }
        default:
            throw new pgm2.InvalidDayException();
        }
        this.year = year;
    }
}

public class pgm2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("\n\t\t\t\tEnter current Day = \n\n");
        int day = scanner.nextInt();
        System.out.println("\n\t\t\t\tEnter current Month = \n\n");
        int month = scanner.nextInt();
        System.out.println("\n\t\t\t\tEnter current Year = \n\n");
        int year = scanner.nextInt();
        currentDate cobj = new currentDate(day, month, year);
        try {
            cobj.createDate(day, month, year);
            System.out.println("\n\t\t\t\tCurrent Date is =" + cobj.date + " / " + cobj.month + " / " + cobj.year);
        } catch (InvalidDayException | InvalidMonthException e) {
            System.out.println(e.getMessage());
        }
    }

    static class InvalidDayException extends Exception {
        public String getMessage() {
            return "invalid day";
        }
    }

    class InvalidMonthException extends Exception {
        public String getMessage() {
            return "Invalid Month";
        }
    }

    public static class InvalidDateException extends Throwable {

    }
}
```

OUTPUT :



```
student@lplab-Lenovo-Product: ~/190905514/week5
                                Popped Element is = e

                                Popped Element is = d

                                Popped Element is = c
a b
student@lplab-Lenovo-Product:~/190905514/week5$ javac pgm2.java
student@lplab-Lenovo-Product:~/190905514/week5$ java pgm2

                                Enter current Day =


5


                                Enter current Month =


5


                                Enter current Year =


2000


                                Current Date is =5 / 5 / 2000
student@lplab-Lenovo-Product:~/190905514/week5$ 
```

3.Design a Student class with appropriate data members as in Lab 5. Provide your own exceptions namely Seats Filled Exception, which is thrown when Student registration number is >XX25 (where XX is last two digits of the year of joining) Show the usage of this exception handling using Student objects in the main. (Note: Registration number must be a unique number).

pgm3.java

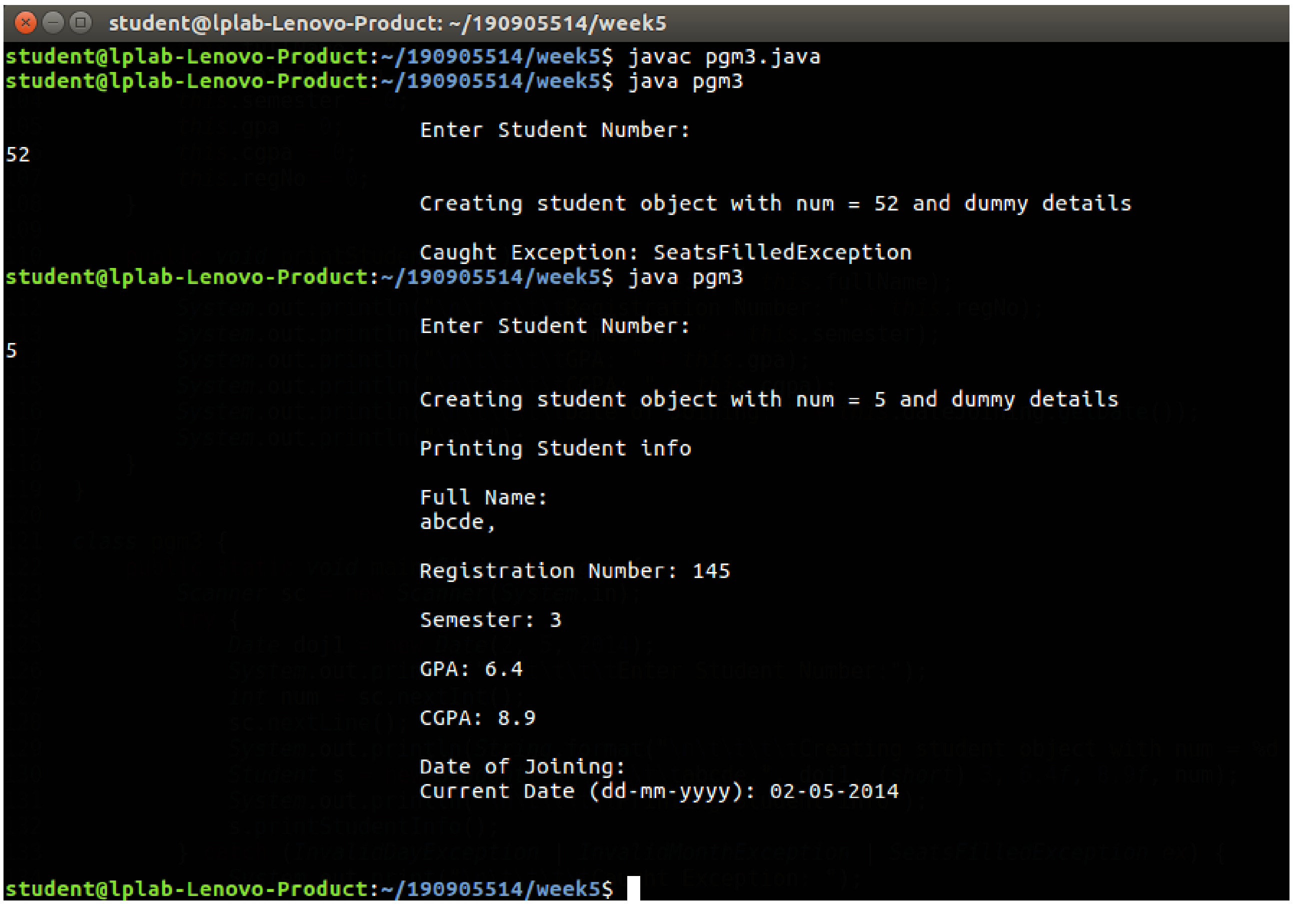
```
import java.util.Scanner;
class InvalidDayException extends Exception {
        int code;

        public InvalidDayException(int c) {
                code = c;
```

```
	}

	public int getCode() {
		return code;
	}
}

class InvalidMonthException extends Exception {
	int code;

	public InvalidMonthException(int c) {
		code = c;
	}

	public int getCode() {
		return code;
	}
}

class SeatsFilledException extends Exception {
	int code;

	public SeatsFilledException(int c) {
		code = c;
	}

	public int getCode() {
		return code;
	}
}

class Date {
	int day, month, year;

	public Date() {
		this.day = 1;
		this.month = 1;
		this.year = 1991;
	}

	public Date(int day, int month, int year) throws InvalidDayException, InvalidMonthException {
		if (month > 12 || month < 1)
			throw new InvalidMonthException(month - 12);
		if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 ||
month == 12) {
			if (day > 31 || day < 1)
				throw new InvalidDayException(day - 31);
		}
		if (month == 4 || month == 6 || month == 9 || month == 11) {
			if (day > 30 || day < 1)
				throw new InvalidDayException(day - 30);
		}
		if (month == 2) {
			if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
				if (day > 29 || day < 1)
					throw new InvalidDayException(day - 29);
			} else {
				System.out.println(day);
				if (day > 28 || day < 1)
					throw new InvalidDayException(day - 28);
			}
		}
		this.day = day;
```

```
			this.month = month;
			this.year = year;
		}

		public String getDate() {
			return (String.format("\n\t\t\t\tCurrent Date (dd-mm-yyyy): %02d-%02d-%04d", this.day,
this.month, this.year));
		}
}

class Student {
		private int regNo;
		private String fullName;
		private Date dateJoining;
		private short semester;
		private float gpa;
		private float cgpa;

		public Student(String fullName, Date dateJoining, short semester, float gpa, float cgpa, int num)
					throws SeatsFilledException {
			if (num > 25)
					throw new SeatsFilledException(num);
			this.fullName = fullName;
			this.dateJoining = dateJoining;
			this.semester = semester;
			this.gpa = gpa;
			this.cgpa = cgpa;
			String reg_year = String.format("%04d", this.dateJoining.year);
			String reg = reg_year.substring(2, 4) + String.format("%s", num);
			this.regNo = Integer.parseInt(reg);
		}

		public Student() {
			this.fullName = "";
			this.dateJoining = new Date();
			this.semester = 0;
			this.gpa = 0;
			this.cgpa = 0;
			this.regNo = 0;
		}

		public void printStudentInfo() {
			System.out.println("\n\t\t\t\tFull Name: " + this.fullName);
			System.out.println("\n\t\t\t\tRegistration Number: " + this.regNo);
			System.out.println("\n\t\t\t\tSemester: " + this.semester);
			System.out.println("\n\t\t\t\tGPA: " + this.gpa);
			System.out.println("\n\t\t\t\tCGPA: " + this.cgpa);
			System.out.println("\n\t\t\t\tDate of Joining: " + this.dateJoining.getDate());
			System.out.println("\n\n");
		}
}

class pgm3 {
		public static void main(String[] args) {
			Scanner sc = new Scanner(System.in);
			try {
					Date doj1 = new Date(2, 5, 2014);
					System.out.println("\n\t\t\t\tEnter Student Number:");
					int num = sc.nextInt();
					sc.nextLine();
					System.out.println(String.format("\n\t\t\t\tCreating student object with num = %d
and dummy details", num));
					Student s = new Student("\n\t\t\t\tabcde,", doj1, (short) 3, 6.4f, 8.9f, num);
```

```
                        System.out.println("\n\t\t\t\tPrinting Student info");
                        s.printStudentInfo();
                } catch (InvalidDayException | InvalidMonthException | SeatsFilledException ex) {
                        System.out.print("\n\t\t\t\tCaught Exception: ");
                        System.out.println(ex);
                }
        }
}
```

OUTPUT :



```
student@lplab-Lenovo-Product:~/190905514/week5$ javac pgm3.java
student@lplab-Lenovo-Product:~/190905514/week5$ java pgm3

                                Enter Student Number:
52

                                Creating student object with num = 52 and dummy details

                                Caught Exception: SeatsFilledException
student@lplab-Lenovo-Product:~/190905514/week5$ java pgm3

                                Enter Student Number:
5

                                Creating student object with num = 5 and dummy details

                                Printing Student info

                                Full Name:
                                abcde,

                                Registration Number: 145

                                Semester: 3

                                GPA: 6.4

                                CGPA: 8.9

                                Date of Joining:
                                Current Date (dd-mm-yyyy): 02-05-2014

student@lplab-Lenovo-Product:~/190905514/week5$
```