/*

* Date: 16-7-25

Your college is conducting a coding seminar limited to 5 students. Students register one after the other. However, if more students register than the limit, the system cannot accept new registrations. Due to frequent cancellations and last-minute changes, students may withdraw their registration, and the system should allow that too.

You are asked to implement the system using a stack, where:

• Push = New student registration (student ID only)

• Pop = Cancel last registration

• Peek = Check who registered last

• Display = Show all current registrations (top to bottom)

Show menu with options:

1. Register student ID (Push)

2. Cancel last registration (Pop)

3. View last registration (Peek)

4. Display all current registrations

5. Exit

When max limit is reached → "Stack overflow! Seminar full."

When trying to cancel from empty list → "Stack underflow! No registrations to remove."

Accurate display of IDs in reverse order of registration (LIFO).*/

```
package julyhometask;
import java.util.*;
class StudentStack
{
	int top;
	int size;
	int[]stu_array;
	public StudentStack(int size)
	{
		this.size=size;
```

```
            stu_array=new int[size];
            top-=1;

        }
        public void insert(int stu_id)
        {
            if(top==size-1)
            {
                System.out.println("The student data is overflown");

            }
            else
            {
                top=top+1;
                stu_array[top]=stu_id;
                System.out.println("The student with "+stu_id+" is registered successfully");

            }
        }
        public void pop()
        {
            if(top==-1)
            {
                System.out.println("The student data is empty");
            }
            else
            {
                System.out.println("The student in "+stu_array[top--]+" is removed from the
student data");
            }
        }
```

```
	public void peek()
	{
		if(top==-1)
		{
			System.out.println("There is no student data");
		}
		else
		{
			System.out.println("The student at top is:" + stu_array[top]);
		}

	}
	public void display()
	{
		if(top==-1)
		{
			System.out.println("There is no student data ");

		}
		else
		{
			System.out.println("The student data is:");
			for (int i=top;i>=0;i--)
			{
				System.out.println(stu_array[i]);
			}
		}
	}

}
```

```
public class july16ht {

	public static void main(String[] args)
	{
		Scanner sc=new Scanner(System.in);
		StudentStack s=new StudentStack(5);
		int choice;
		do
		{
		System.out.println("1.Insert\n2.Cancell\n3.Top student\n4.Display\n5.Exit");
		System.out.println("Enter any option");
		choice=sc.nextInt();
		sc.nextLine();
		switch(choice)
		{
		case 1:

		System.out.println("Enter 5 Student details only");

		for(int i=0;i<5;i++)
		{
			int stu_id=sc.nextInt();
			s.insert(stu_id);
		}
		break;
		case 2:

		s.pop();
		break;
		case 3:
		s.peek();
```

```
                break;
                case 4:
                s.display();
                break;
                case 5:
                System.out.println("--Exiting--");
                break;



        }
                }while (choice!=5);
        }
}
```