

# Data Structure and Algorithm Practicum

## Stack



**Name**

Muhammad Baihaqi Aulia Asy'ari

**NIM**

2241720145

**Class**

1I

**Department**

Information Technology

**Study Program**

D4 Informatics Engineering

---

## 1.1 Learning Objective

After finishing this practicum session, students will be able to:

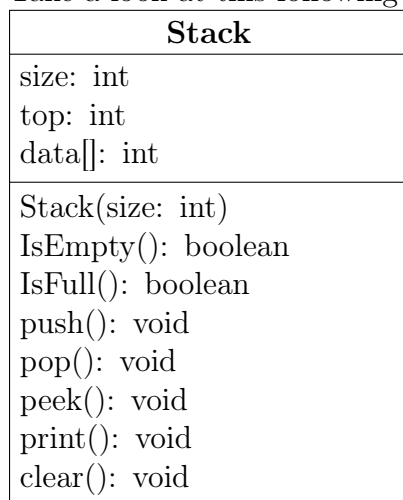
- Define the Stack Data Structure
- Create and implement Stack Data Structure
- Implement Stack data Structure with arrays

## 1.2 Lab Activities

In this practicum, we will implement **Stack** class

### 1.2.1 Steps

1. Take a look at this following class diagram for **Stack** class:



Based on class diagram above, we will create the **Stack** class in Java program.

2. Create a new project named **Jobsheet7**. Create a new package with name **Practicum1**. Then, create a new class named **Stack**.
3. Create new attributes size, top, and data as follows:

```
int size;  
int top;  
int data[];
```

4. Add a constructor with parameter as written below:

```
public Stack(int size) {  
    this.size = size;  
}
```

---

```
        data = new int[size];
        top = -1;
    }
```

5. Create a method **isEmpty** with Boolean as its return type to check whether the stack is empty or not.

```
public boolean isEmpty() {
    if (top == 1) {
        return true;
    } else {
        return false;
    }
}
```

6. Create a method **isFull** with Boolean as its return type to check whether the stack is filled completely or not.

```
public boolean isFull() {
    if (top == size - 1) {
        return true;
    } else {
        return false;
    }
}
```

7. Create method **push** with void as its return type to add new stack element with parameter **dt**. This dt variable is in form of integer

```
public void push(int dt) {
    if (!isFull()) {
        top++;
        data[top] = dt;
    } else {
        System.out.println("Stack is full");
    }
}
```

8. Create method **pop** with void as its return type to remove an element from the stack

```
public void pop() {
    if (!isEmpty()) {
        int x = data[top];
        top--;
    }
}
```

---

```
        System.out.println("Remove data : " + x);
    } else {
        System.out.println("Stack is empty");
    }
}
```

9. Create method **peek** with void as its return type to check the top element of the stack

```
public void peek() {
    System.out.println("Top element : " + data[top]);
}
```

10. Create method **print** with void as its return type to display the content of the stack

```
public void print() {
    System.out.println("Stack content: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i] + " ");
    }
    System.out.println("");
}
```

11. Create method **clear** with void as its data type to remove all elements and make the stack empty

```
public void clear() {
    if (!isEmpty()) {
        for (int i = top; i >= 0; i--) {
            top--;
        }
        System.out.println("Stack is now empty");
    } else {
        System.out.println("Failed ! Stack is still empty");
    }
}
```

12. Next up, we create a new class named **StackMain** inside the package **Practicum1**. Create a main function and make object instantiation with name is **stk**

```
public class StackMain {
    public static void main(String[] args) {
        Stack stk = new Stack(5);
    }
}
```

---

```
    }  
}
```

13. Fill the stack object by calling method **push**, the data is being inserted accordingly

```
stk.push(15);  
stk.push(27);  
stk.push(13);
```

14. Display the data that we've inserted in previous step by calling method **print**

```
stk.print();
```

15. Repeat the insertion process twice, then call pop **method** to remove an element. We can also check the top data with **peek** method. Finally, display all the data by calling method **print**

```
stk.push(11);  
stk.push(34);  
stk.pop();  
stk.peek();  
stk.print();
```

16. Compile and run the program, check the result

```
package Practicum1;  
  
public class Stack {  
    int size;  
    int top;  
    int data[];  
  
    public Stack(int size) {  
        this.size = size;  
        data = new int[size];  
        top = -1;  
    }  
  
    public boolean isEmpty() {  
        if (top == 1) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```

---

```
}

public boolean isFull() {
    if (top == size - 1) {
        return true;
    } else {
        return false;
    }
}

public void push(int dt) {
    if (!isFull()) {
        top++;
        data[top] = dt;
    } else {
        System.out.println("Stack is full");
    }
}

public void pop() {
    if (!isEmpty()) {
        int x = data[top];
        top--;
        System.out.println("Remove data : " + x);
    } else {
        System.out.println("Stack is empty");
    }
}

public void peek() {
    System.out.println("Top element : " + data[top]);
}

public void print() {
    System.out.println("Stack content: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i] + " ");
    }
    System.out.println("");
}

public void clear() {
```

---

```

        if (!isEmpty()) {
            for (int i = top; i >= 0; i--) {
                top--;
            }
            System.out.println("Stack is now empty");
        } else {
            System.out.println("Failed ! Stack is still empty");
        }
    }
}

package Practicum1;

public class StackMain {
    public static void main(String[] args) {
        Stack stk = new Stack(5);

        stk.push(15);
        stk.push(27);
        stk.push(13);

        stk.print();

        stk.push(11);
        stk.push(34);
        stk.pop();
        stk.peek();
        stk.print();
    }
}

```

### 1.2.2 Result

```

1 PS D:\Kuliah> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
   ↪ '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
   ↪ 'C:\Users\ASUS\AppData\Roaming\Code\User\workspaceStorage\
   ↪ ce3fcb236261368a6cbd019dc8ddda8b\redhat.java\
   ↪ jdt_ws\Kuliah_28156aa7\bin' 'Practicum1.StackMain'
2 Stack content:
3 13
4 27
5 15
6

```

---

```
7 Remove data : 34
8 Top element : 11
9 Stack content:
10 11
11 13
12 27
13 15
```

### 1.2.3 Questions

1. In class **StackMain**, what is the usage of number 5 in this following code?  

```
Stack stk = new Stack(5);
```
2. Add 2 more data in the stack with 18 and 40. Display the result!
3. In previous number, the data inserted in to the stack is only 18 and 40 is not inserted. Why is that?

## 1.3 2<sup>nd</sup> Lab Activities

In this practicum, we will create a program to illustrate a bunch of books that are stored in Stack. Since the book has some information on it, the stack implementation is done using array of object to represent each element.

### 1.3.1 Steps

1. This class diagram is used for creating a program code written in Java programming language

| Book  |
|---|
| title: String<br>authorName: String<br>publishedYear: int<br>pagesAmount: int<br>price: int |
| Book(title: String, author: String, publishedYear: int, pagesAmount: int, price: int)       |

2. Create a new package named **Practicum2**, then create a new class named **Book**.
3. Add attributes in that class, and add the constructor as well.



---

```

String title, authorName;
int publishedYear, pageAmount, price;

public Book(String tt, String nm, int yr, int pam, int pr) {
    this.title = tt;
    this.authorName = nm;
    this.publishedYear = yr;
    this.pageAmount = pam;
    this.price = pr;
}

```

4. Copy the program code for Stack class in **Practicum1** to be used again in here. Since the data stored in Stack in **Practicum1** is integer array, and in **Practicum2** we use objects, we will need to modify some parts in that class.
5. Modify the Stack class by changing the data type of **int data[]** to **Book data[]**. This time we will need to save the data in stack in objects. In addition, we will need to change the **attributes**, **constructor**, **method push**, and **method pop**

```

int size, top;
Book data[];

public Stack(int size) {
    this.size = size;
    data = new Book[size];
    top = -1;
}

public void push(Book dt) {
    if (!isFull()) {
        top++;
        data[top] = dt;
    } else {
        System.out.println("Stack is full");
    }
}

```

6. We will need to change the **print**, **pop**, and **peek method** as well since the data that are going to be printed is not only a string, but an object consists of some information (title, authorName, etc.).

```

public void pop() {
    if (!isEmpty()) {

```

---

```

        Book x = data[top];
        top--;
        System.out.println("Remove data : " + x.title + " " +
            ↳ x.authorName + " " + x.publishedYear + " " +
            ↳ x.publishedYear + " " + x.pageAmount + " " +
            ↳ x.price);
    } else {
        System.out.println("Stack is empty");
    }
}

public void peek() {
    System.out.println("Top element : " + data[top]);
}

public void print() {
    System.out.println("Stack content: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i].title + " " +
            ↳ data[i].authorName + " " + data[i].publishedYear + "
            ↳ " + data[i].pageAmount + " " + data[i].price);
    }
    System.out.println("");
}

```

7. Next, we have to create a new class called **StackMain** in **Practicum2**. Create a main function and instantiate an object with named **st**
8. Declare the **Scanner** object with name **sc**
9. Insert these lines of codes to receive **Book** data input, alongside with its information to be stored in stack

```

Stack st = new Stack(8);
Scanner sc = new Scanner(System.in);

char choose;
do {
    System.out.print("Title : ");
    String title = sc.nextLine();

    System.out.print("Author Name : ");
    String name = sc.nextLine();

```

---

```

        System.out.print("Published year : ");
        int year = sc.nextInt();

        System.out.print("Pages Amount : ");
        int pages = sc.nextInt();

        System.out.print("Price : ");
        int price = sc.nextInt();

        Book bk = new Book(title, name, year, pages, price);
        System.out.print("Do you want to add new data to Stack (y/n)?
        ↪ ");
        choose = sc.next().charAt(0);
        sc.nextLine();
        st.push(bk);

    } while (choose == 'y');

```

10. Call print, pop, and peek method accordingly as follows:

```

st.print();
st.pop();
st.peek();
st.print();

```

11. Compile and run **StackMain**, and observe the result

```

package Practicum2;

public class Book {
    String title, authorName;
    int publishedYear, pageAmount, price;

    public Book(String tt, String nm, int yr, int pam, int pr) {
        this.title = tt;
        this.authorName = nm;
        this.publishedYear = yr;
        this.pageAmount = pam;
        this.price = pr;
    }
}

package Practicum2;

```

---

```
public class Stack {
    int size, top;
    Book data[];

    public Stack(int size) {
        this.size = size;
        data = new Book[size];
        top = -1;
    }

    public boolean isEmpty() {
        if (top == -1) {
            return true;
        } else {
            return false;
        }
    }

    public boolean isFull() {
        if (top == size - 1) {
            return true;
        } else {
            return false;
        }
    }

    public void push(Book dt) {
        if (!isFull()) {
            top++;
            data[top] = dt;
        } else {
            System.out.println("Stack is full");
        }
    }

    public void pop() {
        if (!isEmpty()) {
            Book x = data[top];
            top--;
        }
    }
}
```

---

```

        System.out.println("Remove data : " + x.title + " " +
        ↪ x.authorName + " " + x.publishedYear + " " +
        ↪ x.publishedYear + " " + x.pageAmount + " " +
        ↪ x.price);
    } else {
        System.out.println("Stack is empty");
    }
}

public void peek() {
    System.out.println("Top element : " + data[top]);
}

public void print() {
    System.out.println("Stack content: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i].title + " " +
        ↪ data[i].authorName + " " + data[i].publishedYear
        ↪ + " " + data[i].pageAmount + " " +
        ↪ data[i].price);
    }
    System.out.println("");
}

public void clear() {
    if (!isEmpty()) {
        for (int i = top; i >= 0; i--) {
            top--;
        }
        System.out.println("Stack is now empty");
    } else {
        System.out.println("Failed ! Stack is still empty");
    }
}

}

package Practicum2;

import java.util.Scanner;

public class StackMain {
    public static void main(String[] args) {

```

---

```

Stack st = new Stack(8);
Scanner sc = new Scanner(System.in);

char choose;
do {
    System.out.print("Title : ");
    String title = sc.nextLine();

    System.out.print("Author Name : ");
    String name = sc.nextLine();

    System.out.print("Published year : ");
    int year = sc.nextInt();

    System.out.print("Pages Amount : ");
    int pages = sc.nextInt();

    System.out.print("Price : ");
    int price = sc.nextInt();

    Book bk = new Book(title, name, year, pages, price);
    System.out.print("Do you want to add new data to
    ↪ Stack (y/n)? ");
    choose = sc.next().charAt(0);
    sc.nextLine();
    st.push(bk);

} while (choose == 'y');

st.print();
st.pop();
st.peek();
st.print();

sc.close();
}
}

```

### 1.3.2 Result