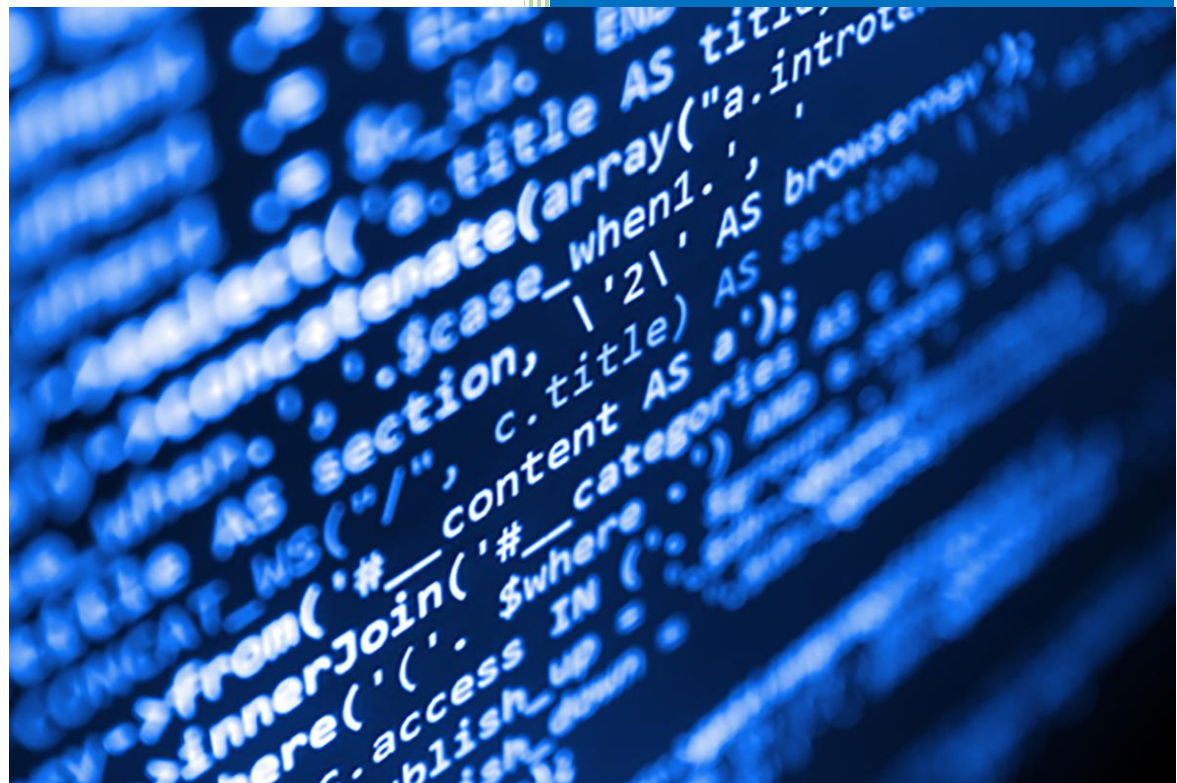




FAKULTI TEKNOLOGI  
KEJURUTERAAN KELAUTAN  
DAN INFORMATIK

2022/2023

# DATA STRUCTURE & ALGORITHM



**Name:** OMAR ISMAIL ABDJALEEL  
ALOMORY  
**Matric Number:** S63955  
**Lab:** MP3  
**Date:** 19/11/2022

## Lab 3: Stack

## TABLE OF CONTENTS

INSTRUCTIONS.....	1
TASK 1: Apply and test the simple implementation of stack .....	2
TASK 2: Decimal to Binary Converter .....	6
TASK 3: Binary To Decimal Converter.....	8

## INSTRUCTIONS

Manual makmal ini adalah untuk kegunaan pelajar-pelajar Fakulti Teknologi Kejuruteraan Kelautan dan Informatik, Universiti Malaysia Terengganu (UMT) sahaja. Tidak dibenarkan mencetak dan mengedar manual ini tanpa kebenaran rasmi daripada penulis.

Sila ikuti langkah demi langkah sebagaimana yang dinyatakan di dalam manual.

This laboratory manual is for use by the students of the Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu (UMT) only. It is not permissible to print and distribute this manual without the official authorisation of the author.

Please follow step by step as described in the manual.

## TASK 1: APPLY AND TEST THE SIMPLE IMPLEMENTATION OF STACK

### OBJECTIVE

In this task, students must be able to:

- Apply the simple implementation of stack.
- Test the implementation

### ESTIMATED TIME

[60 Minutes]

### STEPS:

1. Open Netbeans and create new java application project.
2. Named your project as `StackExperiment` and click finish.
3. Change author profiles to :
  - a. Name :
  - b. Program: <put your program. Eg: SMSK(SE) or SMSK with IM
  - c. Course : CSF3104
  - d. Lab : <enter lab number>
  - e. Date : <enter lab date> [Step 2]
4. Add the following class to your `StackExperiment.java` file after author profiles and before `public class StackExperiment` statement. [Step 3]

```
/** Java implementation of stack of type char */
class ChrStack {

    //1. declare default stack size
    private static final int defaultSize = 10;

    //2. Declare variable for
    private int size;                // Size of stack
    private int top;                 // Index for top Object
    private char[] stack;            // Array holding integer data
}
```

```

/** Constructors */
//3. Constructor to Create stack using default size
ChrStack() {
    size = defaultSize;           // easier version:
    top = -1;                     // replace all 3 lines with
    stack = new char[defaultSize]; // this(defaultSize);
}

//4. Constructor to create stack using supplied size
ChrStack(int size) {
    top = -1;
    this.size = size;
    stack = new char[size];
}

//5. Push method: used to insert data into a top of stack
void push(char data) {
    top++;                       // go to upper position
    stack[top]=data;            // put data in that position
}

//6. Pop method: to get and remove top element/data in stack
public char pop() {
    char data = stack[top] //get data at current
    top--;                 //position and go down one
    return data;           //step and returned the data
}

//7. isempty method: used to check if the stack is empty. Usually used
// before calls to pop method. Returned true or false
boolean isempty() {
    return(top == -1); //returned true if stack is empty
}

```

```

//8. isfull method: used to check if the stack is full. Usually used
// before calls to push. Returned true or false
boolean isfull() {
    return(top == size-1); //returned true if stack is full
}

//9. Length method: to get current top value
int length(){
    return top;
}

} //end of class

```

5. Put the following code in your main method.

```

//1. declare a stack s using above Stack class with size 100
ChrStack s = new ChrStack(100); // 100 chars

//2. Declare a character ch to store character input.
char ch;

//3. Give an instruction to user
System.out.println("Please enter your name:");

//4. Get input and store into stack using push method until user
    press ENTER (\n)
while ((ch = (char) System.in.read()) != '\n') {
    if (!s.isFull()) { //check stack is not full
        s.push(ch); //put data into stack
    }
}

//5. Tell users, their name will be print reversely
System.out.println("Your name reverse is:");
while (!s.isEmpty()) { // check stack is not empty

```

```

        System.out.print(s.pop()); // get data from stack
                                   // using pop method
    }

```

## QUESTIONS

1. What did the program do?

**Answer:** *the program did printed whatever I entered, save it into the stack[] then reversed it while popping the stack and printing the return values.*

2. If your input is "Fakulti Teknologi Kejuruteraan Kelautan Dan Informatik", what is inside stack s during you enter for word "Teknologi"?

**Answer:** *Teknologi is just a string of characters saved in the array stack[], from index 8 to 16.*

3. Try put the following input and see what happen. Explain why it happened.

Do not get worried and do not get scared, we are fighting to get there. So remember, out there somewhere you have got a friend and you will never walk alone again

**Answer:** *it stopped at 'somew', the reason is that we initialized the stack size to hold 100 chars, and what is beyond it will not be saved in the stack.*

4. If the stack is intended to store the integer data, what should be change in a ChrStack above (will be use later in Task2)?

**Answer:** *in CharStack class we would change the stack from char to int type, along with some method like push (we would change the parameter or argument to int type too), and the pop method we would change the return type to int.*

## TASK 2: DECIMAL TO BINARY CONVERTER

### OBJECTIVE

In this task, students must be able to use stack to store data for converting decimal number to binary number.

### ESTIMATED TIME

[60 Minutes]

### STEPS:

1. Create a new project on java applications in netbeans.
2. Named the projects as `Dec2Bin`
3. Beside the default java class that have main method, add new file of java class to your project and named the file as `IntStack`.
4. Copy the contents of `ChrStack` in Task 1 into `IntStack` file and change accordingly so it can support integer data.
5. Go to main method of `Decimal2Binary.java` file and do the following
  - a. Declare a new stack `s` with size 25 using `IntStack` class.
  - b. Declare a variable `decNumber` of type integer.
  - c. Instantiate input function based on `Scanner` class. Named the object as `scanInput`. (You need to import `Scanner` class before doing this).
  - d. Get an integer input and store in a `decNumber`.
  - e. Define a boolean variable called `stop` with default value `false`.
  - f. Repeat the following code until `stop` is set to `true`.
    - i. Push the remainder of `decNumber` divided by 2 into stack  

```
if (!s.isFull())  
    s.push(decNumber % 2)
```
    - ii. Get the quotient of division and store back into `decNumber`  

```
decNumber = decNumber / 2
```
    - iii. Check if we can stop the loop. The loop must be stop when current value of `decNumber` is either 0 or 1. The loop is stopped when variable `stop` is `true`. Before stop, push the final value of `decNumber` into stack `s`

```
if (decNumber == 0 || decNumber == 1){
```



```

        if (!s.isFull())
            s.push(decNumber);
        stop = true;} }

```

6. The stack `s` now contains the binary value of decimal number. Now is time to pop it out one-by-one onto screen.
7. Repeat the following code until the stack is empty.

```

while(!s.isEmpty()){
    System.out.print(s.pop()+" ");
}

```

---

## QUESTIONS

1. In Step 5d, user may enter non-integer input, to handle that situation you may use try-catch block. Show how try-catch block can be implemented in above program

```

while (check) {
    try {
        System.out.println("Please Enter a Number:");
        decNumber = Integer.parseInt(scanInput.nextLine());
        check = false;
    } catch (NumberFormatException n) {
        System.out.println("-----\n"
            + "This input is invaild!!!\n"
            + "-----");
    }
}

```

Answer:

## TASK 3: BINARY TO DECIMAL CONVERTER

### ESTIMATED TIME

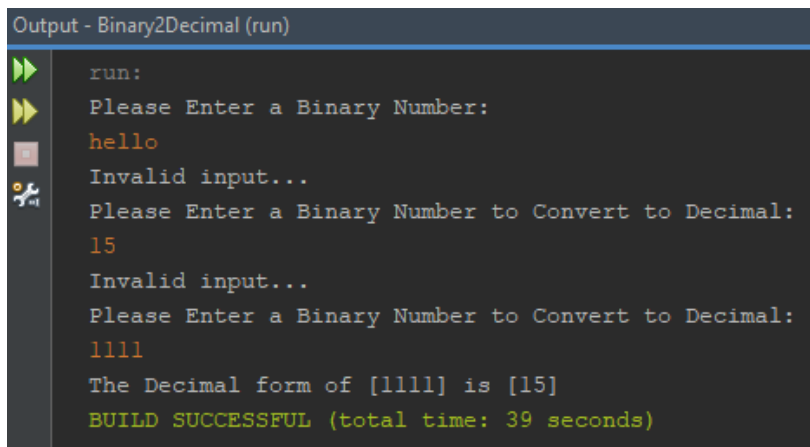
[60 Minutes]

#### STEPS:

Using a similar step in Task 2, write a java program to convert binary number to decimal number. The given algorithm may help you to get the idea.

```
Algorithm Binary2Decimal
Input: number contains only 1 or 0
Output: Decimal number of that binary number
Begin
    1. Read the input
    2. Check and make sure the input contains only 1 and 0
    3. If 2 is correct then put the input into stack, otherwise error
       message or exceptions handler
    4. Get the length of stack (l)
    5. Repeat until the stack is empty
        a. Get the length of stack (t)
        b. Pop the stack (v)
        c. answer = answer +  $(2^{(l-t)} \times v)$ 
    6. Display the answer
End
```

Example of output:



```
Output - Binary2Decimal (run)
run:
Please Enter a Binary Number:
hello
Invalid input...
Please Enter a Binary Number to Convert to Decimal:
15
Invalid input...
Please Enter a Binary Number to Convert to Decimal:
1111
The Decimal form of [1111] is [15]
BUILD SUCCESSFUL (total time: 39 seconds)
```

**INSTRUCTION:** Please submit this lab module and zip all the codes (only java files) to **epembelajaran**.