

CENTRE FOR

DIPLOMA PROGRAMME

MULTIMEDIA UNIVERSITY®

**DCS5068**

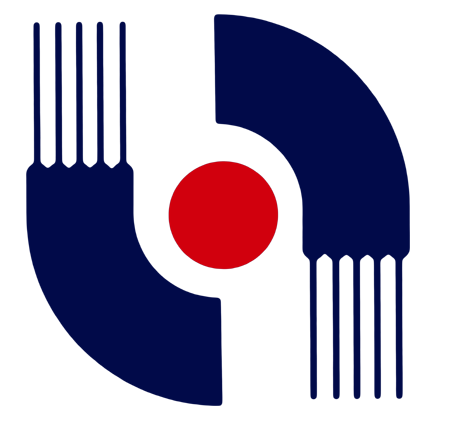
**DATA STRUCTURE AND ALGORITHMS**

*Trimester 1 2014-2015*

GROUP PROJECT REPORT

**Assignment 1**

BY



|  |  |  |
| --- | --- | --- |
| **No.** | **Student ID** | **Student Name** |
| 1. | 1121117048 | NAZRIN PUTRA BIN RASOL |
| 2. | 1121116476 | NISA SABRINA BT. MD. ALASHARI |

GROUP SECTION: DM 1-2

List of Structure Name, Function Name and Class Name

1. Structure Name

* struct Student
* struct node

1. Function Name

* void DisplaySearch(Student s)
* void DisplayQueue(Student student)
* void DisplayStack(Student student)
* void DisplayStatusDelete(int del)

1. Stack method

* int empty()
* void push(Student num)
* Student pop()

1. Queue method

* int empty()
* void append(Student num)
* Student serve()

1. Class Name

* class ADTstack
* class ADTqueue

Source code

#include <iostream>

#include <cstring>

**using** **namespace** std**;**

struct Student

**{**

char name**[**50**],** id**[**10**],** prog**[**3**];**

**};**

struct node

**{**

Student data**;**

node **\***next**;**

**};**

class ADTstack

**{**

private**:**

node **\***top**;**

public**:**

ADTstack**()**

**{**

top **=** **NULL;**

**}**

int empty**()**

**{**

**if(**top **==** **NULL)**

**return** 1**;**

**else**

**return** 0**;**

**}**

void push**(**Student num**)**

**{**

node **\***temp**;**

temp **=** **new** node**;**

temp**->**data **=** num**;**

**if(**top **==** **NULL)**

**{**

top **=** temp**;**

temp**->**next **=** **NULL;**

**}**

**else**

**{**

temp**->**next **=** top**;**

top **=** temp**;**

**}**

**}**

Student pop**()**

**{**

Student num**;**

node **\***temp**;**

**if(!**empty**())**

**{**

num **=** top**->**data**;**

temp **=** top**;**

top **=** top**->**next**;**

**delete** temp**;**

**return** num**;**

**}**

**else**

**{**

cout**<<**"Stack is Empty"**<<**endl**;**

**}**

**}**

**};**

class ADTqueue

**{**

private**:**

node **\***front**,** **\***rear**;**

public**:**

ADTqueue**()**

**{**

front **=** **NULL;**

rear **=** **NULL;**

**}**

int empty**()**

**{**

**if(**front **==** **NULL)**

**return** 1**;**

**else**

**return** 0**;**

**}**

void append**(**Student num**)**

**{**

**if(**rear **!=** **NULL)**

**{**

rear**->**next **=** **new** node**;**

rear **=** rear**->**next**;**

rear**->**data **=** num**;**

rear**->**next**=NULL;**

**}**

**else**

**{**

front **=** rear **=** **new** node**;**

front**->**data**=**num**;**

front**->**next**=NULL;**

**}**

**}**

Student serve**()**

**{**

Student num**;**

**if(!**empty**())**

**{**

num **=** front**->**data**;**

node **\***temp **=** front**;**

front **=** front**->**next**;**

**delete** temp**;**

**if** **(**front **==** **NULL)**

rear **=** **NULL;**

**return** num**;**

**}**

**else**

**{**

cout**<<**"Queue is Empty"**;**

**}**

**}**

**};**

void DisplaySearch**(**Student s**)**

**{**

cout**<<**endl**<<**"Student Name: "**<<**s**.**name**;**

cout**<<**endl**<<**"Student ID: "**<<**s**.**id**;**

cout**<<**endl**<<**"Programme: "**<<**s**.**prog**<<**endl**;**

**}**

void DisplayQueue**(**Student student**)**

**{**

cout**<<**endl**<<**"Student Name: "**<<**student**.**name**;**

cout**<<**endl**<<**"Student ID: "**<<**student**.**id**;**

cout**<<**endl**<<**"Programme: "**<<**student**.**prog**<<**endl**;**

**}**

void DisplayStack**(**Student student**)**

**{**

cout**<<**endl**<<**"Student Name: "**<<**student**.**name**;**

cout**<<**endl**<<**"Student ID: "**<<**student**.**id**;**

cout**<<**endl**<<**"Programme: "**<<**student**.**prog**<<**endl**;**

**}**

void DisplayStatusDelete**(**int del**)**

**{**

**if(**del**==**0**)**

**{**

cout**<<**"\nStatus: Error. Student does not exist."**<<**endl**;**

**}**

**else** **if(**del**==**1**)**

**{**

cout**<<**"\nStatus: Data deleted."**<<**endl**;**

**}**

**}**

int main**()**

**{**

Student s**,** student**,** search**,** remove**,** display**;**

ADTstack stack**;**

ADTqueue queue**;**

node **\***head**,** **\***temp**,** **\***start**,** **\***prev**;**

char delid**[**10**],** searchid**[**10**];**

int selmain**,** sellinked**,** selenter**,** seldisplay**,** data**,** add**,** i**,** found**,** del**;**

cout**<<**"Welcome to Data Structure and Algorithm.\nPlease select your option\n\n\t1. Linked List\n\t2. Exit\n\nSelection:"**;**

cin**>>**selmain**;**

cout**<<**"\n-------------------------------------------------------------------------\n"**;**

**if(**selmain**==**1**)**

**{**

cout**<<**"Selection : Linked List\n\n\t1. Insert new Data\n\t0. Exit\n\nSelection :"**;**

cin**>>**sellinked**;**

cout**<<**"\n-------------------------------------------------------------------------\n"**;**

**if(**sellinked**==**1**)**

**{**

cout**<<**"Selection : 1 - Insert new Node\n\nHow many do you want to insert? :"**;**

cin**>>**data**;**

cout**<<**endl**<<**"Student Name: "**;**

cin**.**ignore**();**

cin**.**getline**(**s**.**name**,**50**);**

cout**<<**"Student ID : "**;**

cin**>>**s**.**id**;**

cout**<<**"Programme : "**;**

cin**>>**s**.**prog**;**

head **=** **new** node**;**

head**->**data **=** s**;**

head**->**next **=** **NULL;**

**for(**i**=**0**;**i**<(**data**-**1**);**i**++)**

**{**

cout**<<**endl**<<**"Student Name: "**;**

cin**.**ignore**();**

cin**.**getline**(**s**.**name**,**50**);**

cout**<<**"Student ID : "**;**

cin**>>**s**.**id**;**

cout**<<**"Programme : "**;**

cin**>>**s**.**prog**;**

temp**=new** node**;**

temp**->**data **=** s**;**

temp**->**next **=** head**;**

head **=** temp**;**

**}**

cout**<<**endl**<<**"Enter : \n\n\t2. Delete Data\n\t3. Search Data\n\t4. Add new Data ( existing list )\n\t5. Display Information\n\t0. Exit\n\nSelection :"**;**

cin**>>**selenter**;**

cout**<<**"\n-------------------------------------------------------------------------\n"**;**

**while(**selenter**!=**0**)**

**{**

**if(**selenter**==**2**)**

**{**

cout**<<**"Selection : 2 - Delete Data\n\nEnter student ID to be deleted\n\nStudent ID:"**;**

cin**>>**delid**;**

del**=**0**;**

start**=**head**;**

**while(**start**!=NULL)**

**{**

remove**=**start**->**data**;**

**if(**strcmp**(**remove**.**id**,** delid**)** **==** 0**)**

**{**

del**=**1**;**

prev**=**start**;**

**break;**

**}**

start**=**start**->**next**;**

**}**

DisplayStatusDelete**(**del**);**

**if(**del**==**1**)**

**{**

**delete** prev**;**

**}**

cout**<<**endl**<<**"Enter : \n\n\t2. Delete Data\n\t3. Search Data\n\t4. Add new Data ( existing list )\n\t5. Display Information\n\t0. Exit\n\nSelection :"**;**

cin**>>**selenter**;**

cout**<<**"\n-------------------------------------------------------------------------\n"**;**

**}**

**else** **if(**selenter**==**3**)**

**{**

cout**<<**"Selection : 3 - Search Data\n\nEnter student ID to search\n\nStudent ID: "**;**

cin**>>**searchid**;**

found**=**0**;**

start**=**head**;**

**while(**start**!=NULL)**

**{**

search**=**start**->**data**;**

**if(**strcmp**(**search**.**id**,** searchid**)** **==** 0**)**

**{**

found**=**1**;**

**break;**

**}**

start**=**start**->**next**;**

**}**

**if(**found**==**0**)**

**{**

cout**<<**"\nStatus: Error. Student does not exist."**<<**endl**;**

**}**

**else** **if(**found**==**1**)**

**{**

cout**<<**"\nStatus: Student Found"**<<**endl**;**

DisplaySearch**(**search**);**

**}**

cout**<<**endl**<<**"Enter : \n\n\t2. Delete Data\n\t3. Search Data\n\t4. Add new Data ( existing list )\n\t5. Display Information\n\t0. Exit\n\nSelection :"**;**

cin**>>**selenter**;**

cout**<<**"\n-------------------------------------------------------------------------\n"**;**

**}**

**else** **if(**selenter**==**4**)**

**{**

cout**<<**"Selection : 4 - Add New Node\n\nHow many data do you want to add? "**;**

cin**>>**add**;**

**for(**i**=**0**;**i**<**add**;**i**++)**

**{**

cout**<<**endl**<<**"Student Name: "**;**

cin**.**ignore**();**

cin**.**getline**(**s**.**name**,**50**);**

cout**<<**"Student ID : "**;**

cin**>>**s**.**id**;**

cout**<<**"Programme : "**;**

cin**>>**s**.**prog**;**

temp**=new** node**;**

temp**->**data **=** s**;**

temp**->**next **=** head**;**

head **=** temp**;**

**}**

cout**<<**endl**<<**"Enter : \n\n\t2. Delete Data\n\t3. Search Data\n\t4. Add new Data ( existing list )\n\t5. Display Information\n\t0. Exit\n\nSelection :"**;**

cin**>>**selenter**;**

cout**<<**"\n-------------------------------------------------------------------------\n"**;**

**}**

**else** **if(**selenter**==**5**)**

**{**

cout**<<**"Selection : 5 - Display Information\n\n\t1. Queue Method\n\t2. Stack Method\n\nSelection :"**;**

cin**>>**seldisplay**;**

start**=**head**;**

**while(**start**!=NULL)**

**{**

display**=**start**->**data**;**

stack**.**push**(**display**);**

queue**.**append**(**display**);**

start**=**start**->**next**;**

**}**

**if(**seldisplay**==**1**)**

**{**

**while(!**queue**.**empty**())**

**{**

queue**.**serve**();**//empty queue

student**=**stack**.**pop**();**//empty stack

DisplayQueue**(**student**);**

**}**

**}**

**else** **if(**seldisplay**==**2**)**

**{**

**while(!**stack**.**empty**())**

**{**

stack**.**pop**();**//empty stack

student**=**queue**.**serve**();**//empty queue

DisplayStack**(**student**);**

**}**

**}**

cout**<<**endl**<<**"Enter : \n\n\t2. Delete Data\n\t3. Search Data\n\t4. Add new Data ( existing list )\n\t5. Display Information\n\t0. Exit\n\nSelection :"**;**

cin**>>**selenter**;**

cout**<<**"\n-------------------------------------------------------------------------\n"**;**

**}**

**else**

**{**

**return** 0**;**//exit

**}**

**}**

**}**

**else**

**{**

**return** 0**;**//exit

**}**

**}**

**else**

**{**

**return** 0**;**//exit

**}**

cout**<<**"Selection : 0 - Exit\n\n-------------------------------------------------\n\n\t\tTHANK YOU\n\n-------------------------------------------------"**;**

**return** 0**;**//exit

**}**