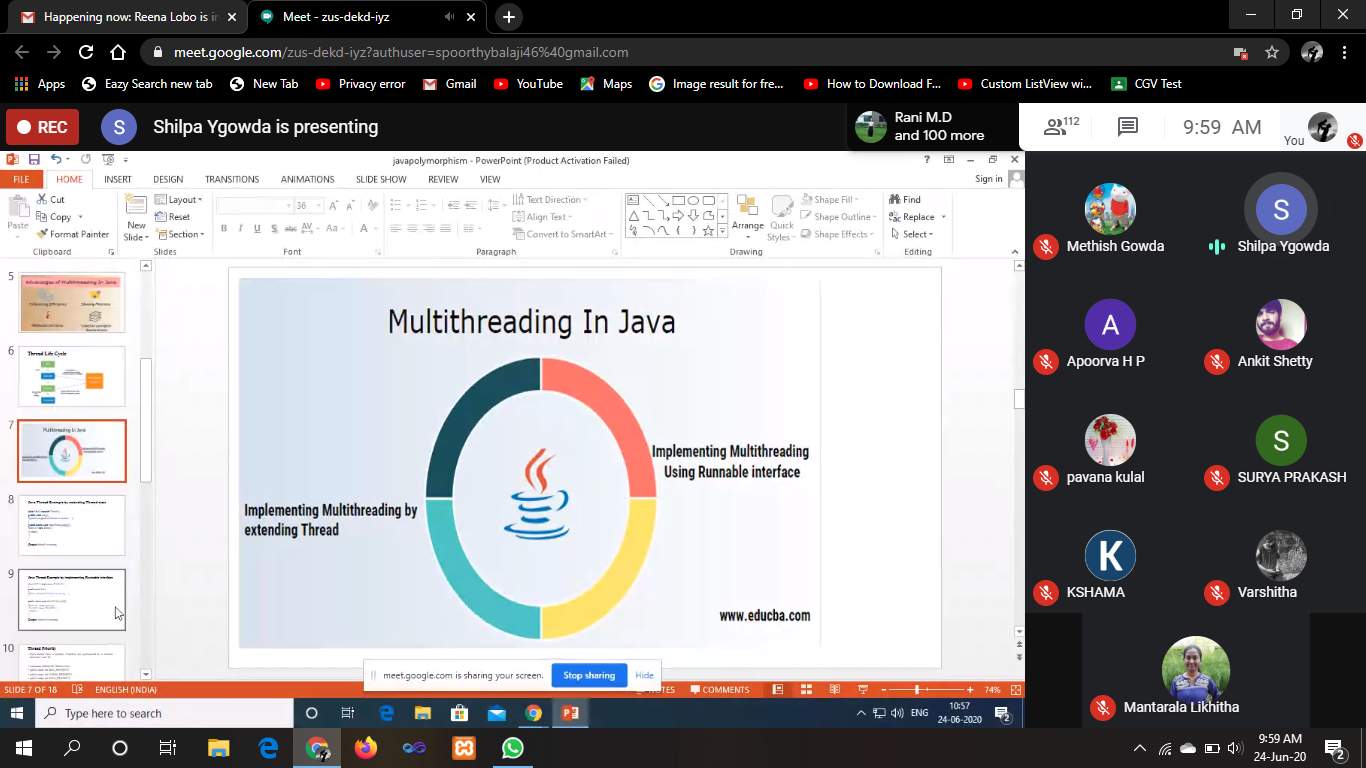
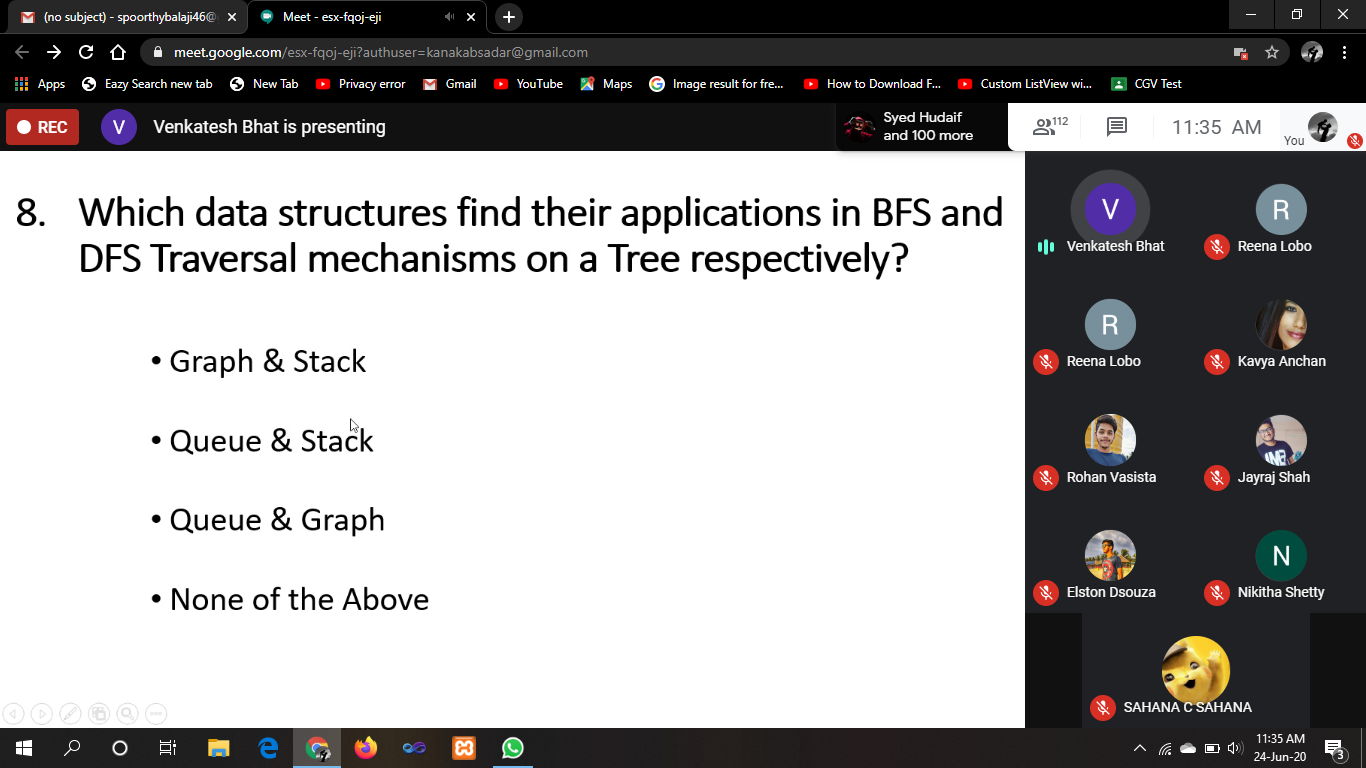
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | | **24/06/2020** | | | | **Name:** | **Spoorthy Balaji** | |
| **Sem & Sec** | | **6th & B** | | | | **USN:** | **4al17cs098** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | | **JAVA & J2EE and Data Structures in C** | | | | | |
| **Max. Marks** | | | **-** | **Score** | | | **-** | |
| **Pre-placement Training Summary** | | | | | | | | |
| **Topic** | **JAVA & J2EE ,**  **Data Structures in C** | | | | | | | |
| **Faculty** | Shilpa  Venkatesh Bhat | | | | **Duration** | | | **4hours** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:** 3 programs | | | | | | | | |
| **Status: Solved** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | <https://github.com/spoorthybalaji/Daily_Status> | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

**SNAPSHOTS**

****

****

ONLINE CODING

**1. Python Program to Create a Class and Compute the Area and the Perimeter of the Circle**

import math

class circle():

def \_\_init\_\_(self,radius):

self.radius=radius

def area(self):

return math.pi\*(self.radius\*\*2)

def perimeter(self):

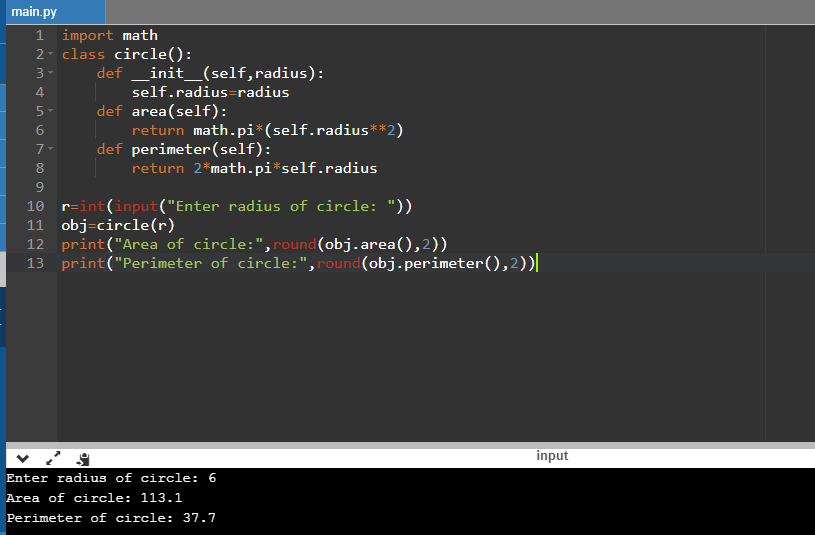
return 2\*math.pi\*self.radius

r=int(input("Enter radius of circle: "))

obj=circle(r)

print("Area of circle:",round(obj.area(),2))

print("Perimeter of circle:",round(obj.perimeter(),2))



**2. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.**

class Shape{

public void print\_shape(){

System.out.println("This is shape");

}

}

class Rectangle extends Shape{

public void print\_rect(){

System.out.println("This is rectangular shape");

}

}

class Circle extends Shape{

public void print\_circle(){

System.out.println("This is circular shape");

}

}

class Square extends Rectangle{

public void print\_square(){

System.out.println("Square is a rectangle");

}

}

public class Test{

public static void main(String[] args){

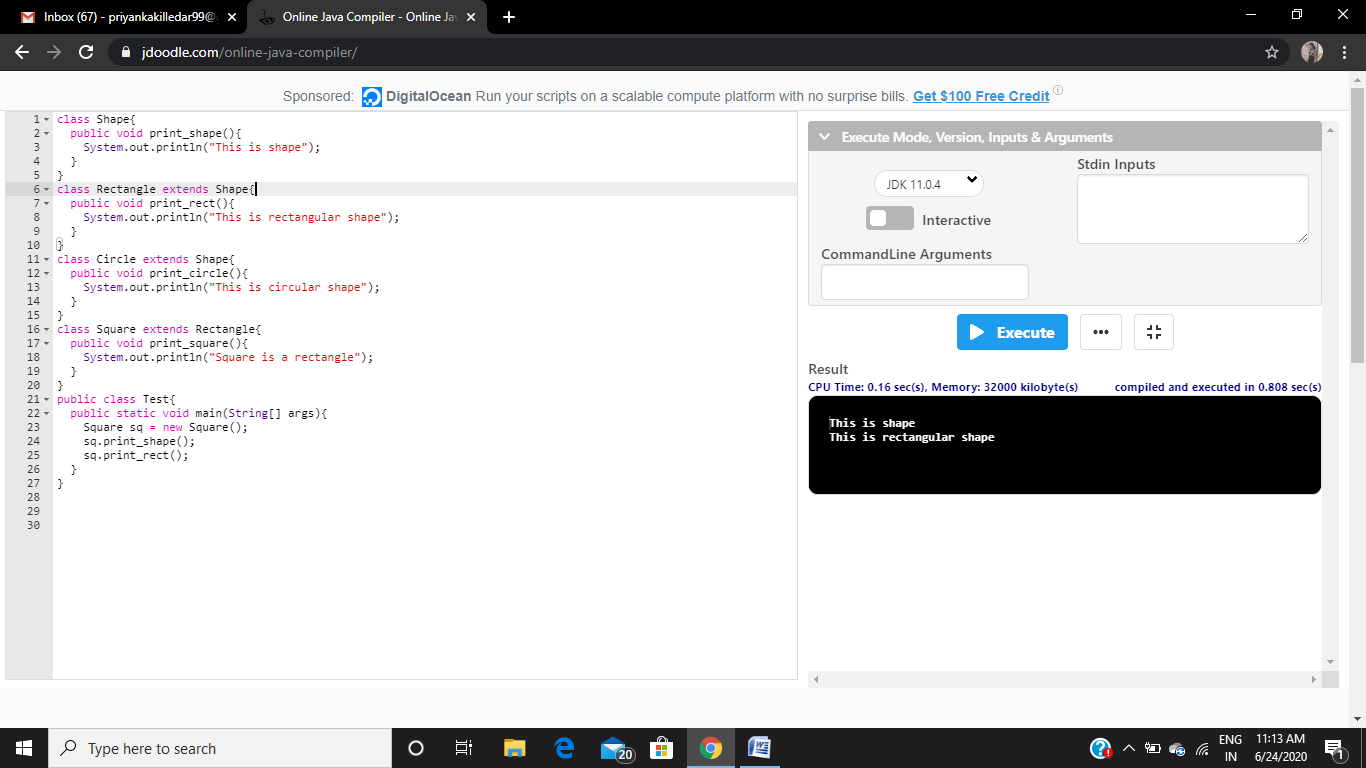
Square sq = new Square();

sq.print\_shape();

sq.print\_rect();

}

}



**3. Sort stack using temporary stack**

#include <stdio.h>

#include <stdlib.h>

struct stack

{

int data;

struct stack \*next;

};

void initStack(struct stack \*\*s)

{

\*s = NULL;

}

int isEmpty(struct stack \*s)

{

if (s == NULL)

return 1;

return 0;

}

void push(struct stack \*\*s, int x)

{

struct stack \*p = (struct stack \*)malloc(sizeof(\*p));

if (p == NULL)

{

fprintf(stderr, "Memory allocation failed.\n");

return;

}

p->data = x;

p->next = \*s;

\*s = p;

}

int pop(struct stack \*\*s)

{

int x;

struct stack \*temp;

x = (\*s)->data;

temp = \*s;

(\*s) = (\*s)->next;

free(temp);

return x;

}

int top(struct stack \*s)

{

return (s->data);

}

void sortedInsert(struct stack \*\*s, int x)

{

if (isEmpty(\*s) || x < top(\*s))

{

push(s, x);

return;

}

int temp = pop(s);

sortedInsert(s, x);

push(s, temp);

}

void sortStack(struct stack \*\*s)

{

if (!isEmpty(\*s))

{

int x = pop(s);

sortStack(s);

sortedInsert(s, x);

}

}

void printStack(struct stack \*s)

{

while (s)

{

printf("%d ", s->data);

s = s->next;

}

printf("\n");

}

int main(void)

{

struct stack \*top;

initStack(&top);

push(&top, 75);

push(&top, 93);

push(&top, 28);

push(&top, 67);

push(&top, 35);

push(&top, 41);

printf("Stack elements before sorting:\n");

printStack(top);

sortStack(&top);

printf("\n\n");

printf("Stack elements after sorting:\n");

printStack(top);

return 0;

}

