ASSIGNMENT 08

1) Write a Java program to read two numbers a and b and calculate a/(a-b). The program should check the value of a-b before dividing with a, it should throw an exception if a-b is zero. In the exception handler the program should display appropriate message to the user.

Protected Access Specifiers

```
class Animal {
    protected void display() {
        System.out.println("I am an animal");
    }
}

class Dog extends Animal {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.display();
    }
}
```

Public Access Specifier

```
class Animal {
    // public variable
    public int legCount;

    // public method
    public void display() {
        System.out.println("I am an animal.");
        System.out.println("I have " + legCount + " legs.");
    }
}

public class Main {
    public static void main( String[] args ) {
        Animal animal = new Animal();
        animal.legCount = 4;
        animal.display();
    }
}
```

OUTPUT:

Protected

```
I am an animal.
```

Public

```
I am an animal.
I have 4 legs.
```

2) Write a Java program to implement Stack using class and object.

```
import java.util.*;
```

```
class Stk{
    public int SIZE = 10;
    public int stack[]=new int[SIZE];
    public int top = -1;

    public boolean isFull(){
        if(top==SIZE-1){
            return true;
        }
        else{
            return false;
        }
    }
    public boolean isEmpty(){
        if(top==-1){
            return true;
        }
        else{
            return false;
        }
    }
    public void push(int x){
        if(isFull()){
            System.out.println("Stack is full");
            System.exit(1);
        }
        else{
            top = top + 1;
            stack[top] = x;
            System.out.println("The inserted element is: "+stack[top]);
        }
    }
    public int pop(){
        int de=0;
        if(isEmpty()){
            System.out.println("Stack is empty");
            System.exit(1);
        }
        else{
            de = stack[top];
            top = top - 1;
        }
        return de;
    }
    public void display(){
        for(int i = 0; i<=top;i++){
            System.out.print(stack[i]+", ");
            //System.out.println();
        }
    }
}
class StackImp{
    public static void main(String args[]){
        int v, d;
        Scanner in = new Scanner(System.in);
        Stk ob = new Stk();
        int ch;
        while(true){
            System.out.println();
            System.out.println("Choose from following options");
            System.out.println("1. Push");
            System.out.println("2. Pop");
            System.out.println("3. Display");
            ch = in.nextInt();
            switch(ch){
                case 1: System.out.println("Enter the value");
                        v = in.nextInt();
                        ob.push(v);
                        break;
                case 2: d = ob.pop();
                        System.out.println("The poped element is: "+d);
                        break;
                case 3: ob.display();
                        break;
                default: System.out.println("Invalid Input");
            }
        }
    }
}
```

OUTPUT:

```
Choose from following options
1. Push
2. Pop
3. Display
1
Enter the value
2
The inserted element is: 2

Choose from following options
1. Push
2. Pop
3. Display
1
Enter the value
3
The inserted element is: 3

Choose from following options
1. Push
2. Pop
3. Display
1
Enter the value
5
```

```
The inserted element is: 5

Choose from following options
1. Push
2. Pop
3. Display
3
2, 3, 5,
Choose from following options
1. Push
2. Pop
3. Display
2
The poped element is: 5

Choose from following options
1. Push
2. Pop
3. Display
3
2, 3,
Choose from following options
1. Push
2. Pop
3. Display
```

3) Write a Java program to implement Linear Queue by using class and object.

```java
import java.util.*;
class Qu{
    public int SIZE = 10;
    public int q[]=new int[SIZE];
    public int front = -1;
    public int rear = -1;

    public boolean isFull(){
        if(rear==SIZE-1){
            return true;
        }
        else{
            return false;
        }
    }
    public boolean isEmpty(){
        if(front==-1 && front == rear +1){
            return true;
        }
        else{
            return false;
        }
    }
    public void insert(int x){
        if(isFull()){
            System.out.println("Queue is full");
            System.exit(1);
        }
        else{
            if(front == -1){
                front = 0;
            }
            rear = rear + 1;
            q[rear] = x;
            System.out.println("The inserted element is: "+q[rear]);
        }
    }
    public int delete(){
        int de=0;
        if(isEmpty()){
            System.out.println("Queue is empty");
            System.exit(1);
        }
        else{
            de = q[front];
            front = front + 1;
        }
        return de;
    }
    public void display(){
        if(isEmpty()){
            System.out.println("Queue is empty");
        }
        System.out.println("Queue is");
        for(int i=front;i<=rear;i++){
            System.out.print(q[i]+", ");
        }
        System.out.println();
    }
}
class QueueImp{
    public static void main(String args[]){
        int v, d;

        Scanner in = new Scanner(System.in);
        Qu ob = new Qu();
        int ch;
        while(true){
            System.out.println();
            System.out.println("Choose from following options");
            System.out.println("1. Insert");
            System.out.println("2. Delete");
            System.out.println("3. Display");
            ch = in.nextInt();
            switch(ch){
                case 1: System.out.println("Enter the value");
                        v = in.nextInt();
                        ob.insert(v);
                        break;
                case 2: d = ob.delete();
```

```
                        System.out.println("The poped element is: "+d);
                        break;
                case 3: ob.display();
                        break;
                default: System.out.println("Invalid Input");
            }
        }
    }
}
```

OUTPUT:

```
Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
2
The inserted element is: 2

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
3
The inserted element is: 3

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
5
The inserted element is: 5

Choose from following options
1. Insert
2. Delete
3. Display
3
Queue is
2, 3, 5,

Choose from following options
1. Insert
2. Delete
3. Display
2
The poped element is: 2
```

```
Choose from following options
1. Insert
2. Delete
3. Display
3
Queue is
3, 5,

Choose from following options
1. Insert
2. Delete
3. Display
```

4) Write a Java program to implement Circular Queue by using class and object.

```java
package folder1;


import java.util.Scanner;

public class Demo {
    public int n1;
    public void getOne(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the 1st number");
        n1 = in.nextInt();
    }
}
```

```java
package folder2;

import folder1.Demo;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int no2;
        Scanner in = new Scanner(System.in);
        Demo ob = new Demo();
        ob.getOne();
        System.out.println("Enter the 2nd number");
        no2 = in.nextInt();
        if(ob.n1>no2){
            System.out.println(ob.n1+" is larger");
        }
        else{
            System.out.println(no2+" is larger");
        }
    }
}
```

OUTPUT:

```
Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
2
The inserted element is: 2

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
3
The inserted element is: 3

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
4
The inserted element is: 4

Choose from following options
1. Insert
2. Delete
3. Display
3
Queue is
2,
3,
4,

Choose from following options
1. Insert
2. Delete
3. Display
2
The poped element is: 2

Choose from following options
1. Insert
2. Delete
3. Display
3
```

```
Queue is
3,
4,

Choose from following options
1. Insert
2. Delete
3. Display
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, GROUP – B2
BRANCH - CSE