//Program to Implement Queue Aswin Asok

```
import java.util.Scanner;
public class Printjob
{
   int job[];  //Data Members
   int Capacity;
   int NewJob;
   int Front;
   int Rear;

   Printjob() //Constructor to initialize the data members.
   {
      Capacity = 20;
      Front = Rear = -1;
      createJob();
   }

   void createJob() //Function to create the initialize queue.
   {
      job = new int[Capacity];
   }

   void addJob() //Function to add a new item to the queue.
   {
      if(Rear == Capacity - 1)
      {
         System.out.println("Printjob is full, cannot add any more");
      }
      else
      {
         if(Rear == -1){
            Front = Rear = 0;
         }
         else
         {
            Rear ++;
```

```
        }
        job[Rear] = NewJob;
    }
}

void removeJob() //Function to remove an item from the queue.
{
    if(Front == -1)
    {
        System.out.println("Printjob is empty");
    }
    else
    {
        if(Front == Rear)
            Front = Rear = -1;
        else
            Front++;
    }
}

void displayJob() //Function to display all the elements in the queue.
{
    if(Front == -1)
        System.out.println("Printjob is empty");

    else
    {
        System.out.println("\nStack Elements");
        for(int i = Front ;i<=Rear;i++)
            System.out.println(job[i]);
    }
}

public static void main(String args[]) // main function to show user the available functions
{
    Scanner br=new Scanner(System.in);
    char ask = 'Y';
```

```
        Printjob p1 = new Printjob(); // Object Creation

        do
        {
           System.out.println("\n-----------Menu----------"); //Menu
           System.out.println("1).Push");
           System.out.println("2).Pop");
           System.out.println("3).Display");
           System.out.println("4).Exit");
           System.out.println("Enter Your Choice");
           int choice = br.nextInt();
           switch(choice)
           {
              case 1: System.out.println("Enter Item to Push");
              p1.NewJob = br.nextInt();
              p1.addJob();
              break;

              case 2: p1.removeJob();
              break;

              case 3: p1.displayJob();
              break;

              case 4:System.exit(0);
              break;

           }
           System.out.println("\nDo you want to Continue(Y/N)?");
           ask = br.next().charAt(0);
        }
        while(ask == 'Y' || ask == 'y');
     }
}
```

OUTPUT

```
----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
3
Printjob is empty

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
1
Enter Item to Push
56

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
1
Enter Item to Push
65
```

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
3

Stack Elements
56
65

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
2

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
3

Stack Elements
65

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
1
Enter Item to Push
99

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop
3).Display
4).Exit
Enter Your Choice
3

Stack Elements
65
99

Do you want to Continue(Y/N)?
Y

----------Menu----------
1).Push
2).Pop

3).Display
4).Exit
Enter Your Choice
4