JAVA IOT DEVELOPER LAB

LAB -2

SUBMITTED BY:

AYUSH KUMAR JHA

SAP ID - 500086400

Enrollment no - R200220083

B.C.A -I.O.T.

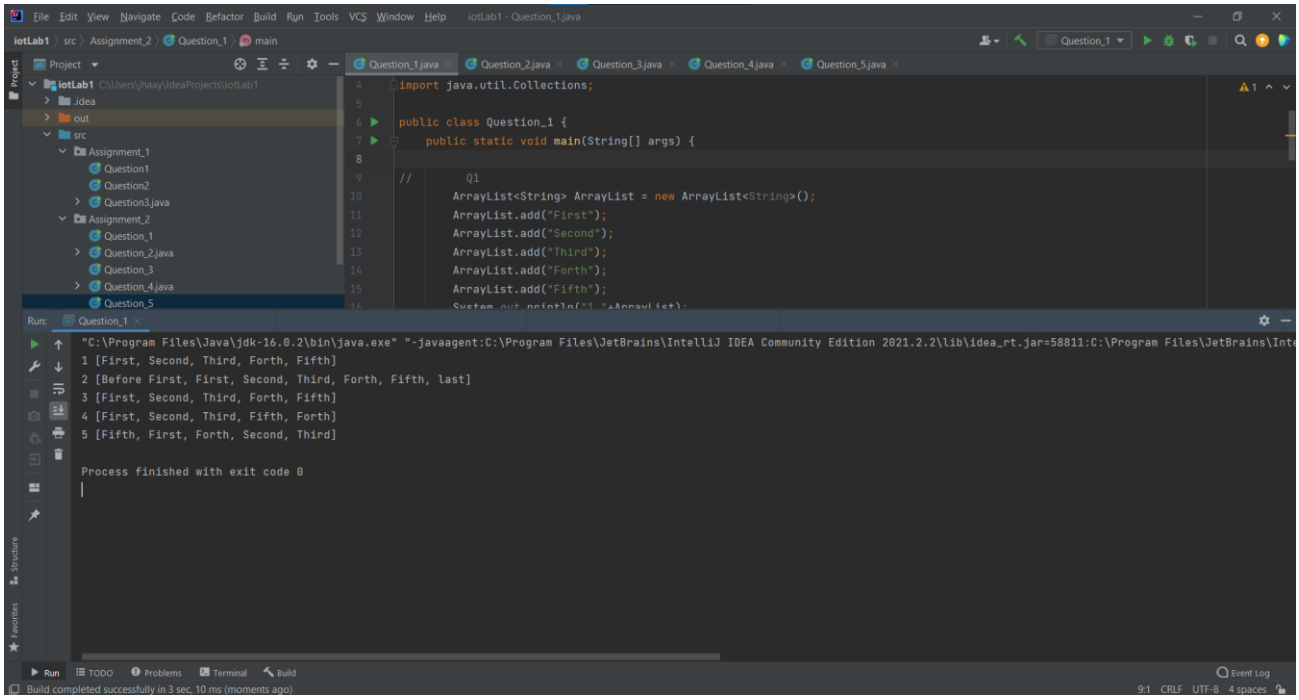
SUBMITTED TO:

SURBHI SARASWAT Ma'am

Questions :-

1. Create a grocery list using ArrayList interface.Initialise it with 5 elements and display the list.Add two more elements (one at start and one at end) and display the listDelete two elements (one by name and one by index value) and display the list. Replace element present at index 3 and get the element present at index 4. Sort the final list.

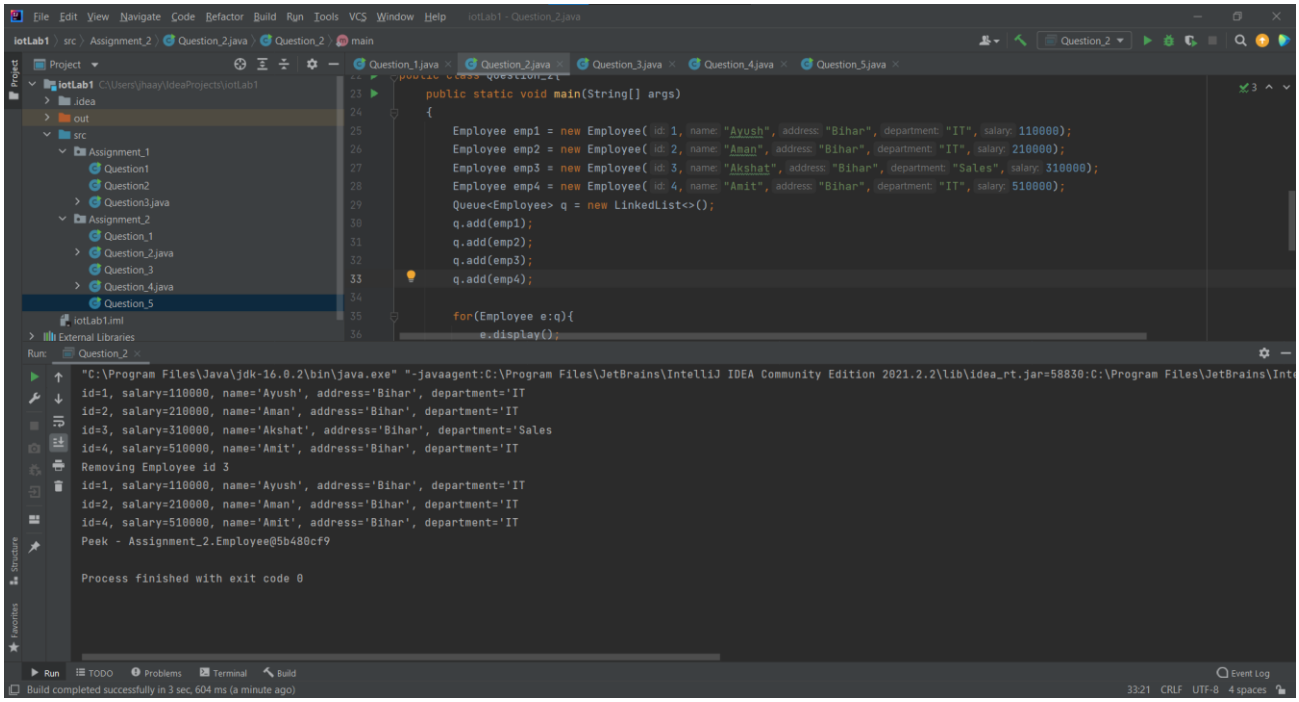
```
import java.util.ArrayList;
import java.util.Collections;

public class Question_1 {
   public static void main(String[] args) {

//      Q1
      ArrayList<String> ArrayList = new ArrayList<String>();
      ArrayList.add("First");
      ArrayList.add("Second");
      ArrayList.add("Third");
      ArrayList.add("Forth");
      ArrayList.add("Fifth");
      System.out.println("1 "+ArrayList);

//      Q2
      ArrayList.add(0,"Before First");
      ArrayList.add("last");

      System.out.println("2 "+ArrayList);

//      Q3
      ArrayList.remove("Before First");
      ArrayList.remove(ArrayList.size()-1);

      System.out.println("3 "+ArrayList);

//      Q4
      String third = ArrayList.get(3);
      ArrayList.set(3,ArrayList.get(4));
      ArrayList.set(4,third);

      System.out.println("4 "+ArrayList);

//      Q5

      Collections.sort(ArrayList);

      System.out.println("5 "+ArrayList);
   }
}
```



2. Create an Employee class: Emplyee(int id, String name, String address, String department, int salary). Then create an employee list using the Queue interface Use methods add, remove, and peek.

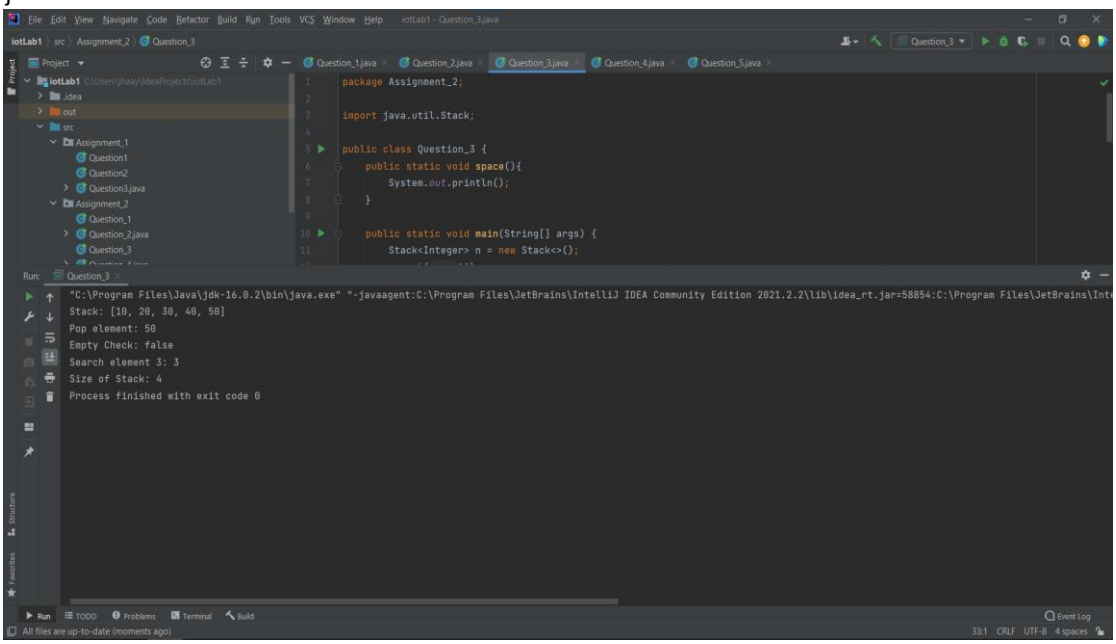
```
import java.util.LinkedList;
import java.util.Queue;

class Employee {
   int id, salary;
   String name, address, department;

   Employee(int id,  String name, String address, String department,int salary) {
      this.id = id;
      this.salary = salary;
      this.name = name;
      this.address = address;
      this.department = department;
   }

   public void display() {
      System.out.println("id=" + id + ", salary=" + salary +", name='" + name + '\'' +", address='" + address
+ '\'' + ", department='" + department);
   }
}
public class Question_2{
   public static void main(String[] args)
   {
      Employee emp1 = new Employee(1,"Ayush","Bihar","IT",110000);
      Employee emp2 = new Employee(2,"Aman","Bihar","IT",210000);
```

```
        Employee emp3 = new Employee(3,"Akshat","Bihar","Sales",310000);
        Employee emp4 = new Employee(4,"Amit","Bihar","IT",510000);
        Queue<Employee> q = new LinkedList<>();
        q.add(emp1);
        q.add(emp2);
        q.add(emp3);
        q.add(emp4);

        for(Employee e:q){
            e.display();
        }

        System.out.println("Removing Employee id 3");
        q.remove(emp3);

        for(Employee e:q){
            e.display();
        }

        System.out.println("Peek - "+ q.peek());

    }
}
```



3. Create a Java Stack interface with int type and illustrate : push, pop, empty, search, and size methods.

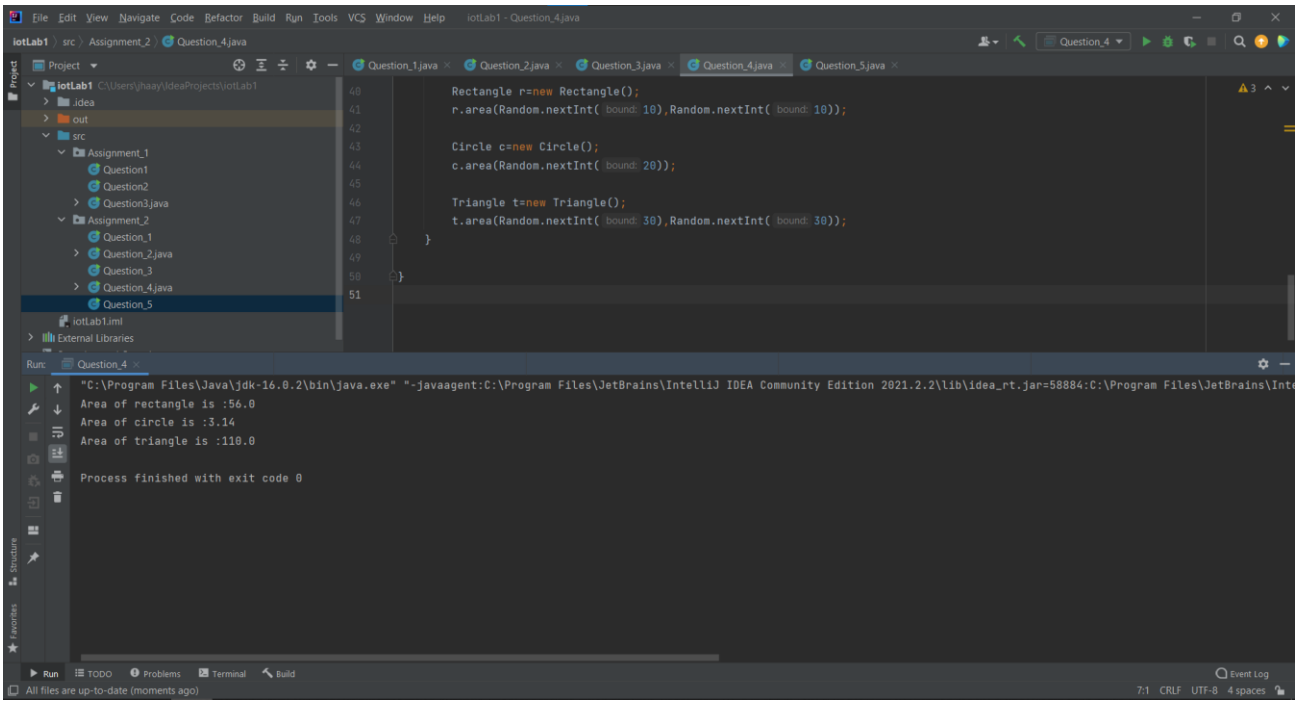
```
import java.util.Stack;

public class Question_3 {
   public static void space(){
      System.out.println();
   }

   public static void main(String[] args) {
      Stack<Integer> n = new Stack<>();
      n.push(10);
      n.push(20);
      n.push(30);
      n.push(40);
      n.push(50);

      System.out.print("Stack: " + n);
      Question_3.space();

      System.out.print("Pop element: " + n.pop());
      Question_3.space();

      System.out.print("Empty Check: " + n.empty());
      space();

      System.out.print("Search element 3: " + n.search(20));
      space();

      System.out.print("Size of Stack: " + n.size());
   }
}
```



4. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

```
import java.util.Random;

abstract class shape
{
   abstract void area(double x,double y);
   int x,y;
}
class Rectangle extends shape
{
   void area(double x,double y)
   {
      System.out.println("Area of rectangle is :"+(x*y));
   }
}
class Circle extends shape
{
   void area(double x,double y)
   {
      System.out.println("Null case");
   }
   void area(double x)
   {
      System.out.println("Area of circle is :"+(3.14*x*x));
   }
}
class Triangle extends shape
{
   void area(double x,double y)
   {
      System.out.println("Area of triangle is :"+(0.5*x*y));
   }
}

public class Question_4 {
   public static void main(String[] args) {
      Random Random = new Random();

      Rectangle r=new Rectangle();
      r.area(Random.nextInt(10),Random.nextInt(10));

      Circle c=new Circle();
      c.area(Random.nextInt(20));

      Triangle t=new Triangle();
```

```
        t.area(Random.nextInt(30),Random.nextInt(30));
    }

}
```



5. Write a program to create two sets A and B of type string using any of the set interface.
   i. Create a set C which is union of set A and set B.
   ii. Verify that set C is a subset of set A.
   iii. Read a new set D from user and find the intersection of the new set and Set C.

```
import java.util.*;

public class Question_5 {
    public static void main(String[] args) {
        Set<String> a = new HashSet<>();
        Set<String> b = new HashSet<>();
        //Adding Elements
        a.add("One");
        a.add("Two");
        a.add("Three");

        System.out.println("Set A: " + a);

        //Adding Elements
```

```
        b.add("Four");
        b.add("Five");
        b.add("Six");
        System.out.println("Set B: " + b);

        //Coping all element from a And b
        Set<String> c = new HashSet<>();
        c.addAll(a);
        c.addAll(b);

        System.out.println("Set c: " + c);
        System.out.println("Set C is subset of set A: " + c.containsAll(a));

        Set<String> d = new HashSet<>();
        Scanner sc = new Scanner(System.in);

        System.out.printf("Enter the Numbers of input:- ");
        int n = sc.nextInt();

        for(int i = 1;i<=n;i++) {
            System.out.printf("Enter the %d one :- ",i);
            String s = sc.next();
            d.add(s);
        }

        Set<String> newSet = new HashSet<>(c);
        newSet.retainAll(d);
        System.out.println("Intersection of element D and c: " + newSet);
    }
}
```