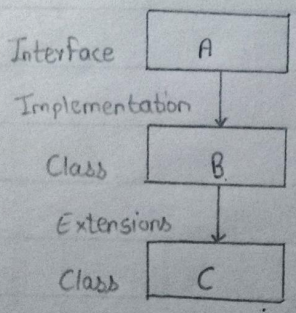
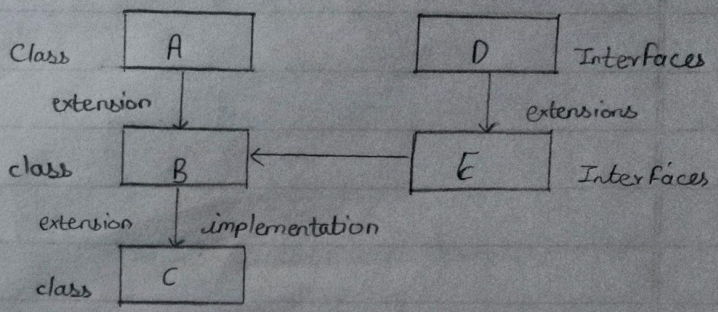
# Practical No. 09

**Aim:** Create, debug and run java programs based on interfaces.

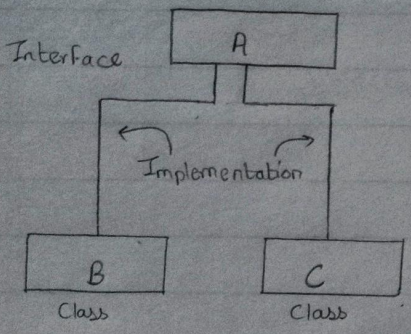
**Diagram:**



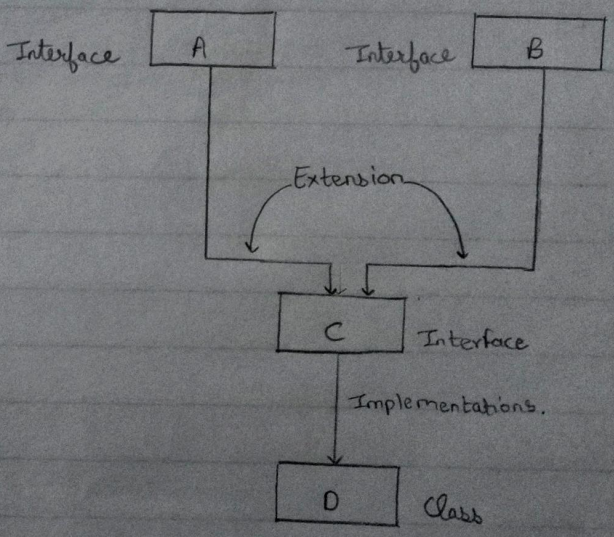

(a)




(b)




(c)




(d)

# Practical No. 9

Aim : Create, debug and run java programs based on interfaces.

Theory :

What is an interface?

→ • An interface is almost like a class with a major difference being that interfaces define only abstract methods and final fields.

- Syntax:

```
interface InterfaceName {
    variable declaration
    Method declaration.
}
```

here, 'interface' is the Keyword, and InterfaceName is any valid Java variable (just like class names).

- Variables are declared as follows:

```
static final type VariableName = Value;
```

example:

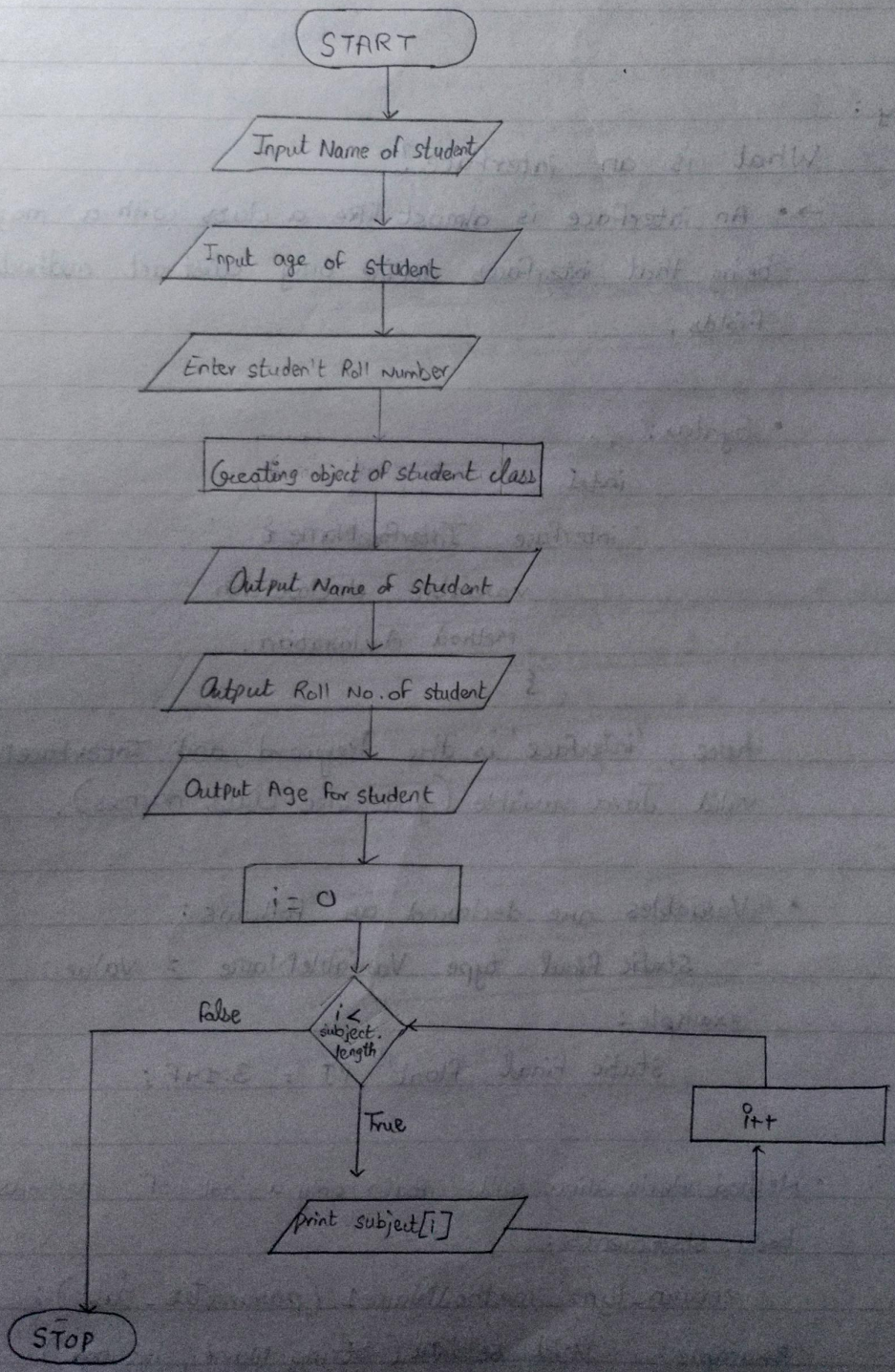
```
Static final float PI = 3.14F;
```

- Method declaration will contain only a list of methods without any body statements:

```
return-type methodName1 (parameter-list);
```

example: `void setData (String Name, int age);`

Flowchart :


START
Input Name of student
Input age of student
Enter student Roll Number
Creating object of student class
Output Name of student
Output Roll No. of student
Output Age for student
i = 0
i < subject.length
False
True
print subject[i]
i++
STOP

## Extending Interfaces:

- An interface can be sub-interfaced from other interfaces.
- The new sub interface will inherit all the members of the sub interface in the manner similar to subclasses.
- This is achieved using the keyword 'extends'.
- Syntax:

```
interface name1 extends name2 {
  // body of name 1
}
```

- We can also combine several interfaces together into a single interface. Syntax:

```
interface a {
  // body of a
}

interface b {
  // body of b
}

interface c extends a, b {
  // body of c
}
```

- Interfaces cannot extend classes as it would violate the rule of interfaces having only final variables and abstract methods.

Conclusion :

Hence, by performing this practical, I learnt about the concept of interfaces and their implementation. I also created, developed, debugged and executed java programs on interfaces

## Code :

```
import java.util.Scanner;

interface Student{
    final String subjects[] = {"English","Maths","Science"};
    void display();
    void showName();
    void showRollNo();
    void showAge();
}

class StudentInfo implements Student{
    String name;
    int rollNo;
    int age;

    StudentInfo(String name, int age, int rollNo){
        this.name = name;
        this.age = age;
        this.rollNo = rollNo;

    }

    public void showName() {
        System.out.println("Name : " + name);
    }

    public void display(){
        showName();
        showRollNo();
        showAge();
        System.out.print("Subjects : ");
        for(String a : subjects){
            System.out.print(a + " ");
        }
    }

    public void showRollNo(){
        System.out.println("Roll No. : " + rollNo);
    }

    public void showAge(){
        System.out.println("Age : "  + age);
    }
}
```

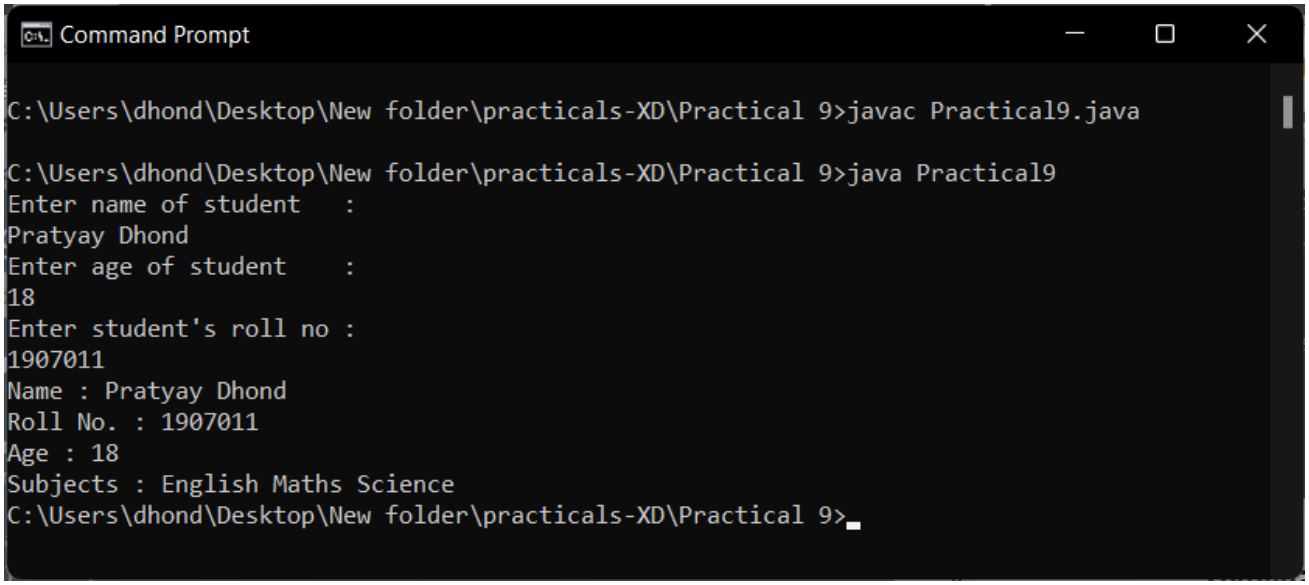
```java
class Practical9{

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String name = " ";
        int age = 0;
        int rollNo = 0;
        System.out.println("Enter name of student   : ");
       name = sc.nextLine();
        System.out.println("Enter age of student    : ");
        age = sc.nextInt();
        System.out.println("Enter student's roll no : ");
        rollNo = sc.nextInt();

        StudentInfo student1 = new StudentInfo(name,age,rollNo);
        student1.display();
    }

}
```

**Output:**



Implementing interfaces :

- Interfaces are used as a "superclass" whose properties are inherited by classes.
- Syntax :

```
class className { implement InterfaceName{
    // body of className
}
```

- A more general / in use form of 'implements' may look like :

```
class className implements extends Superclass
    implements interface1, interface2 .... {
    // body of className
}
```

Thus, a class can extend ~~of~~ another class while implementing interface.

~~Code:~~

Conclusion :

Hence, by performing this practical, I learnt about the concept of interfaces and their implementation. I also created, developed, debugged and executed java programs on interfaces.