Assignment 3

OOP's

Hardik Srivastava

199303069

CCE-3rd-A

1.
```
class Staff {
    int code;
    String name;
    Staff(int code, String name) {
        this.code = code;
        this.name = name;
    }
}
class Teacher extends Staff {
    String subject, publication;
    Teacher(int code, String name, String subject,
            String publication) {
        super(code, name);
        this.subject = subject;
        this.publication = publication;
    }
}
class Officer extends Staff {
    String grade;
    Officer(int code, String name, String grade) {
        super(code, name);
        this.grade = grade;
    }
}
```

```
class Typist extends staff {
    int speed;
    Typist ( int code , string name, int speed){
        super (code , name);
        this . speed = speed;
    }
}

class RegularTypist extends Typist {
    RegularTypist(int code, string name,
                  int speed ){
        super(code , name, speed);
    }
}

class casualTypist{ extends Typist {
    int dailyWages;
   casualTypist(int code, string name, int speed,
                int dailyWages) {
     super(code, name, speed);
     this. dailyWages = daily Wages;
    }
}
```

2.
```
interface A {
    void A( int a, float b);
}

interface B {
    void B( double c, float d);
}

class C implements A, B {
    C ( int a, float b){
        A( a, b);
    }
    C ( double c, float d) {
        B( c, d);
    }

    C ( int a, float b, double c, float d) {
        A( a, b);
        B( c, d);
    }
    @Override
    public void A( int a, float b) {
        System.out.println(a+ "  "+ b);
    }
    @Override
    public void B (double c, float d) {
        System.out.println( c+ "  " + d);
    }
}
```

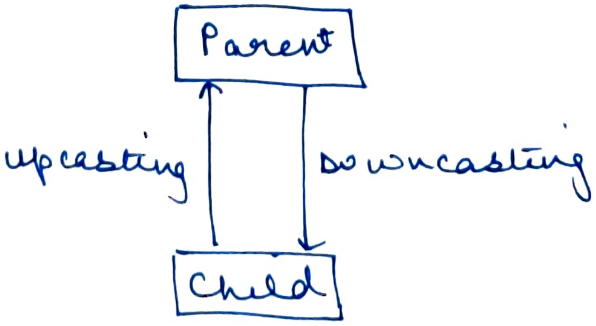
```
class Assignment_3_2 {
    public static void main(String[] args) {
        C c1 = new C(1,2.0F);
        C c2 = new C(5.0, 9.0F);
        C c3 = new C(1,2.0F ,5.0 F,9.0 F);
    }
}
```

3. Upcasting :-

- Upcasting is the typecasting of a child object to a parent object.
- Done Implicitly
- Upcasting gives us the flexibility to access the parent class members but it is not possible to access all the child class members using this feature.

Downcasting :-

- Downcasting means typecasting of a parent object to a child object.
- Cannot be done implicitly.



```
class Parent {
    void method () {
      System.out. println ("Parent method "];
    }
}

class child extends Parent {
    @Override
    void method () {
       System.out.println ("child method "];

    }
}

class Assignment_3_3 {
    public static void main (String [] args) {
        // Upcasting
        Parent p = new Child ();
        p. method ();

        // Downcasting explicitly
        child c = (child) p;
        c. method ();

    }
}
```