EXERCISE-2

Q2

```
public interface TaxCalculator {
      public abstract void hra();
}
public class Humanity implements TaxCalculator {
      private int basic_salary;

      public Order(int basic_salary) {
            this.basic_salary = basic_salary;
      }

      @Override public
      void hra() {
        HRA=(10/100)*basic_salary;

      }
}
public class Logistic implements TaxCalculator {
      private int basic_salary;

      public Order(int basic_salary) {
            this.basic_salary = basic_salary;
      }

      @Override public
      void hra() {
        HRA=(10/100)*basic_salary;
      }
}
public class Department { public static void main(String[]
      args) { basic_salary basic_salary = new
      basic_salary();

            Humanity humanity = new Humanity(basic_salary);
            Logistic logistic = new Logistic(basic_salary);
            Humanity.hra();

            humanity = new humanity(basic_salary);
            logistic = new Logistic(basic_salary);
            Logistic.hra();
      }
}
```

Q3

```
const arr = [1,2,3,4,5,6,7,8,9,10]; const sum =
arr.reduce((acc, val) => acc + val); var
mean=sum/arr.length; console.log("mean
:",mean); const { length: num } = arr; let
variance = 0;
   arr.forEach(arr => {
      variance += ((arr - mean) * (arr - mean));
   }); console.log("variance
",variance/arr.length);
```

Q4

```
class productId
{
constructor( productId, ProductName,Productprice)
{
this.productId=productId;
this.ProductName=ProductName;
this.Productprice=Productprice;
}
}
let ob1=new productId(1,abc,10);
let ob2=new productId(22,def,100);
```