"Polymorphism"

**Employee**
- firstName: String
- lastName: String
- nationalId: String

+ Employee(firstName, lastName, id)
+ earnings(): double
+ getters/setters

**SalariedEmployee**
- weeklySalary: double

+ SalariedEmployee(firstName, lastName, id, salary)
+ earnings(): double
+ getters/setters

**HourlyEmployee**
- wage: double
- hours: double

+ HourlyEmployee(firstName, lastName, id, hourlyWage, hoursWorked)
+ earnings(): double
+ getters/setters

**CommissionEmployee**
- grossSales: double
- commissionRate: double

+ CommissionEmployee(firstName, lastName, id, sales, rate)
+ earnings(): double
+ getters/setters

**BasePlusCommissionEmployee**
- baseSalary: double

+ BasePlusCommissionEmployee(firstName, lastName, id, sales, rate, salary)
+ earnings(): double
+ getters/setters
### Class earnings

<table>
<thead>
<tr>
<th>Class</th>
<th>earnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employee</td>
<td>Abstract</td>
</tr>
<tr>
<td>SalariedEmployee</td>
<td>weeklySalary</td>
</tr>
<tr>
<td>HourlyEmployee</td>
<td>If hours &lt;= 40</td>
</tr>
<tr>
<td></td>
<td>wage * hours</td>
</tr>
<tr>
<td></td>
<td>else if hours &gt; 40</td>
</tr>
<tr>
<td></td>
<td>40 * wage + (hours – 40) * wage * 1.5</td>
</tr>
<tr>
<td>CommissionEmployee</td>
<td>commissionRate * grossSales</td>
</tr>
<tr>
<td>BasePlusCommissionEmployee</td>
<td>(commissionRate * grossSales) + baseSalary</td>
</tr>
</tbody>
</table>

**Tasks to be performed:**

- Implement the classes shown in the UML diagram
- Implement a test class that creates sub-class objects of the abstract Employee class
- The test class will create an Employee array
- Put the above created sub-class objects in this Employee array
- Loop through this employee array and determine the type of the Employee
- Print the earnings of the employees