# Assessment Background/Scenario

The Human Resources (HR) department of a superstore needs an application for managing payroll (calculating employees’ income tax, the wage bill for each department and producing reports). The superstore has three (3) departments as shown in Table 2, and each department has several employees, as shown in Table 1). The application should calculate the level of income tax based on Table 3.

***Table 1:*** *Sample employees’ data.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee ID** | **First Name** | **Last Name** | **Department ID** | **Gross Salary** | **Tax** | **Salary**  **after tax** |
| **1** | Mike | Watts | 1 | £50,000 | £7,486 | £42,514 |
| **2** | Rakesh | Kumar | 2 | £40,000 | £5,486 | £34,514 |
| **3** | Havea | Fonua | 3 | £80,000 | £19,432 | £60,568 |
| **4** | Jack | Moral | 1 | £165,000 | £59,210 | £105,790 |
| **5** | Sarah | Karim | 2 | £12,000 | £0 | £12,000 |
| **6** | Shazia | Naeem | 3 | £38,000 | £5,086 | £32,914 |

***Table 2:*** *Sample departments' data.*

|  |  |  |
| --- | --- | --- |
| **Department ID** | **Name** | **Address** |
| 1 | Human Resources | 1 Willcott Crescent, Mount Albert |
| 2 | Marketing | 45 Pitt Street, Mount Wellington |
| 3 | Production | 30 Willcott Crescent, Mount Albert |

***Table 3:*** *Tax Rate (https://www.gov.uk/income-tax-rates).*

|  |  |  |
| --- | --- | --- |
| **Band** | **Taxable income** | **Tax rate** |
| Personal allowance | Up to £12,570 | 0% |
| Basic rate | £12,571 to £50,270 | 20% |
| Higher rate | £50,271 to £150,000 | 40% |
| Additional rate | over £150,000 | 45% |

# Assessment Tasks

Design and build an application that requires the user to input the payroll data via the console/command line and store it in an appropriate internal structure. There is no requirement for this data to be preserved after the application is closed. You will need to consider what the application outputs to the user to demonstrate the results of each task below. You should provide the user with a means to close the program once operations are complete.

Your application should be able to perform the following tasks given the scenario above:

**Task A:** Write pseudocode algorithms for the tasks given below.

1. Fetch department data (i.e. name and address) and store in an appropriate data structure.

Note: department ID should be auto-incremental as shown in Table 2.

1. Fetch employee data (name, department ID and gross salary) and store in an appropriate data structure.

Notes: (1) employee ID should be auto-incremental (2) application should show appropriate error message if the employee’s department ID entered does not exist.

1. Sort all employees by salary (lowest salary first)
2. Sort all employees by department.
3. Calculate tax for each employee on the salary based on Table 3.
4. Search and identify the employee who has the highest salary as shown in Table 4.

***Table 4:*** *Employee receiving the highest salary.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee ID** | **First Name** | **Last Name** | **Department** | **Gross Salary** | **Tax** | **Salary**  **after tax** |
| **4** | Jack | Moral | Human  Resource | £165,000 | £59,210 | £105,790 |

1. Search and identify the employee who has the lowest salary as shown in Table 5.

***Table 5:*** *Employee receiving the lowest salary.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee ID** | **First Name** | **Last Name** | **Department** | **Gross Salary** | **Tax** | **Salary after tax** |
| **5** | Sarah | Karim | Marketing | £12,000 | £0 | £12,000 |

**Task B:** Construct the application and implement algorithms.

1. Develop a single Java program that enables the user to: store, search, and sort employees.
2. Implement all the algorithms you have created for Task A.
3. Ensure that the one and only one Java file can be compiled and run from the command/console line.

# Deliverables

This work is not submitted unless it is as a requirement of an exam question. You should complete as much of this work as you can during Week 8, as there will be no time during the exam (2 hours, plus 30 minutes for uploads) to do so. You should organise your work to ensure you can access this work during the exam quickly and easily.

# Grading

The pass mark for postgraduate modules is 50. For more information about grades and assessment criteria, please review the ‘Assessment and award’ section of the *York Online Programmes Handbook*, which is available to view or download from your **Orientation Module**.

# Marking Criteria

No marks are awarded directly for the exam pre-work. Where the exam pre-work is referred to, uploaded, or otherwise used in exam questions the marks will be made clear on the examination paper.

# Assessment Policies

This assessment is subject to the policies stated on the ‘Summative Assessment Policies’ page in Canvas. These policies include (but are not limited to):

* Academic Integrity and submission of student work to Turnitin
* Advice on anonymising your assessment
* Penalties for late submission
* Marking policy for multiple submissions
* The Fit to Sit / Submit policy

Please ensure that you have read and understood these policies before starting the   
assessment