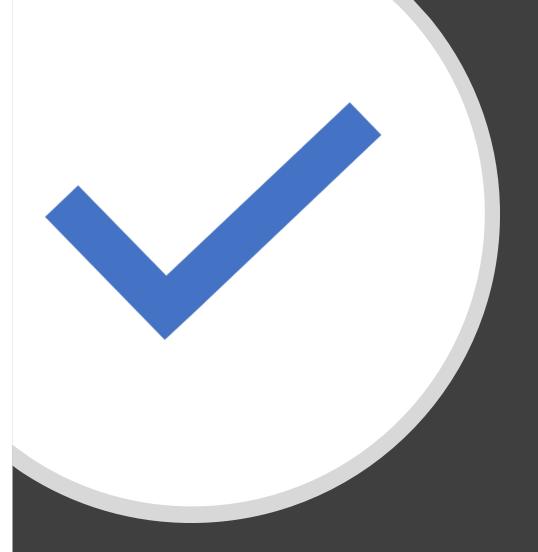


Employee
Wage
Computation
Problem



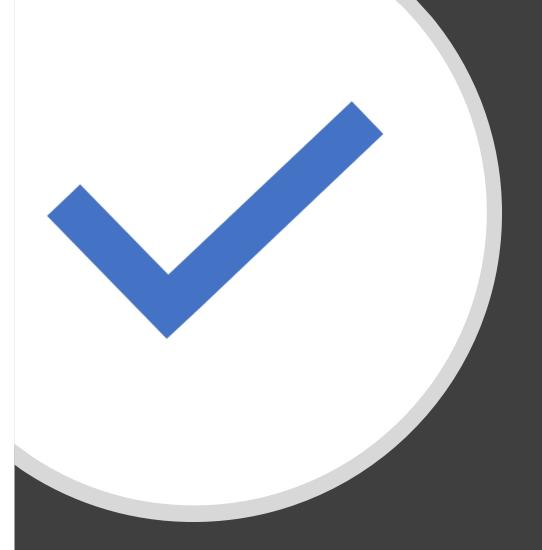
- 1. Ability to work with Bash Shell Scripts Programming Constructs
- 2. Ability to work with basic git flows.





GIT Flow Expectations

- Create Repo for Shell Scripts
- Add and Commit Files
- Follow Message Hygiene
- During Commit specify Add or Refactor in the message beginning
- Create Branch
- Push to Remote Master and Branch
- Merge and Resolve Conflicts
- Ability to look into History of Files
- Comfortable with Git Commands





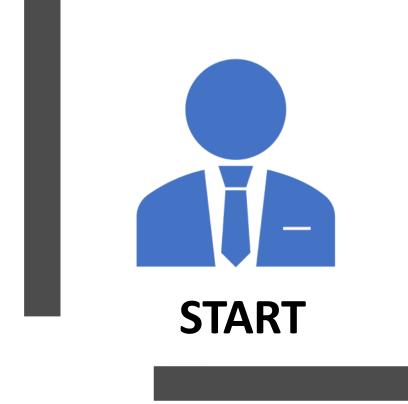
Shell Programming Expectations

- Use of Shell Programming Constructs
- Use of proper names for the File Names,
 Variables and Constants
- Use of Proper Indentations
- Avoiding Printing to standard terminal instead use debug
- No Commented Codes
- Follow DRY Principle Do not Repeat Yourself
- Comfortable with Debug Execution Threads
- Comfortable with Shell Commands

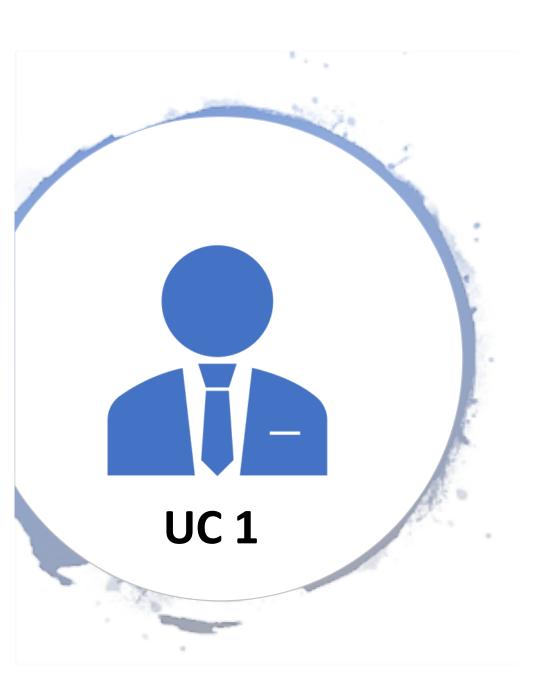
Rules

- Create empWageComputation.sh
- Start with Welcome message in the Main Branch
- Every Use Case (UC) in the Corresponding UC Branch
- For e.g. UC-1 Branch Name EmployeeAttendanceUC
- Follow Programming Hygiene and DRY principle
- Testing the Program before pushing to Remote
- On Completion of every UC, do the following
 - Merge Local Branch with Local Master
 - Push to Remote Branch
 - Merge Local Master with Local Branch
 - Push Local Master to Remote Master



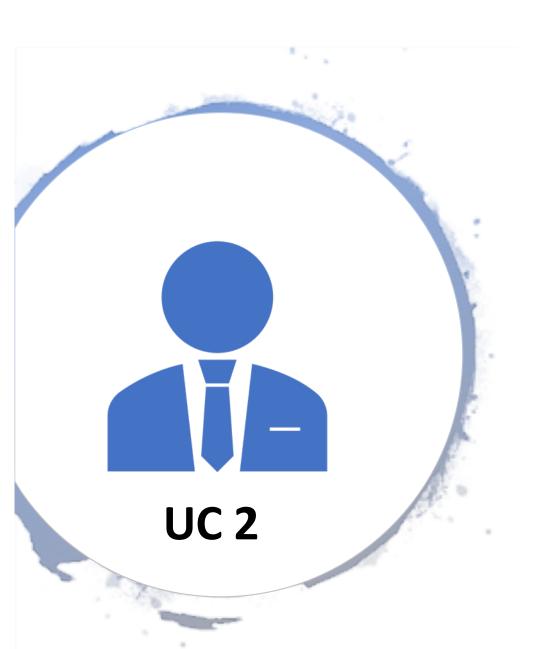


Start with Displaying
Welcome to Employee
Wage Computation
Program on Master Branch



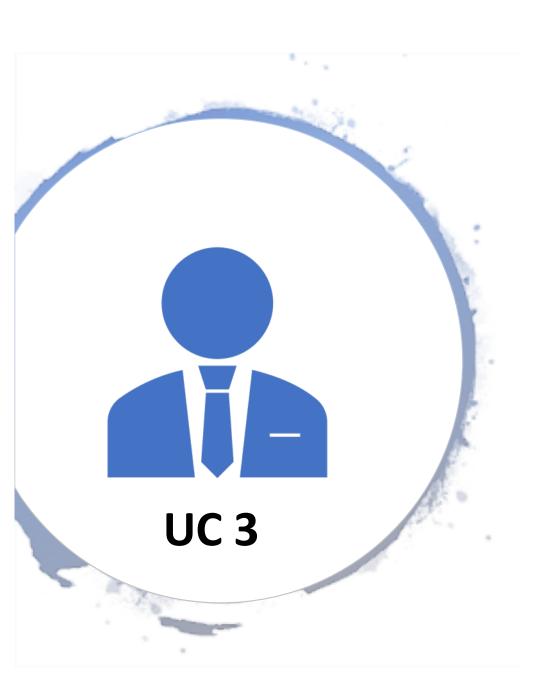
Check Employee is Present or Absent

- Use ((RANDOM)) for Attendance Check



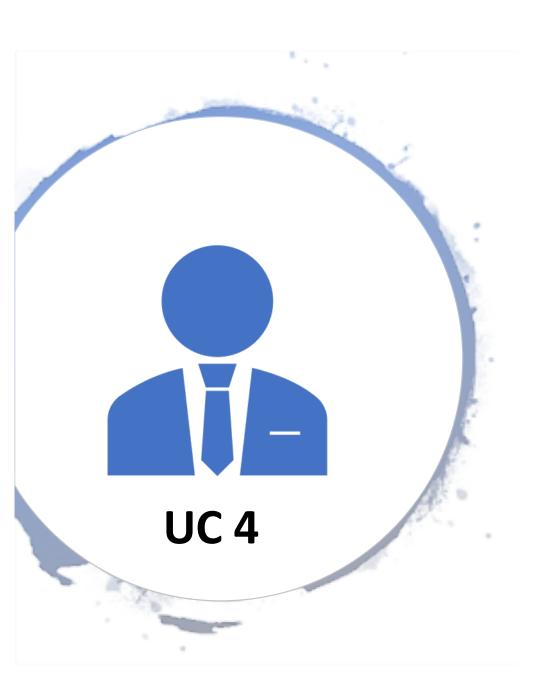
Calculate Daily Employee Wage

- Assume Wage per Hour is 20
- Assume Full Day Hour is 8

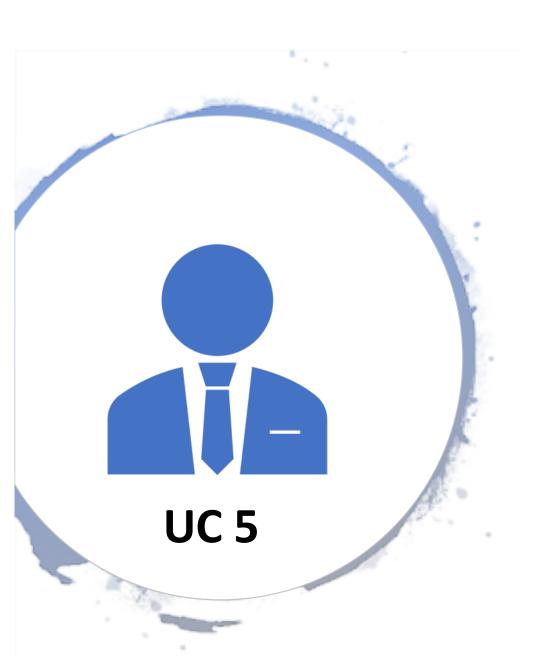


Add Part time Employee & Wage

- Assume Part time Hour is 8



Solving using Switch Case Statement



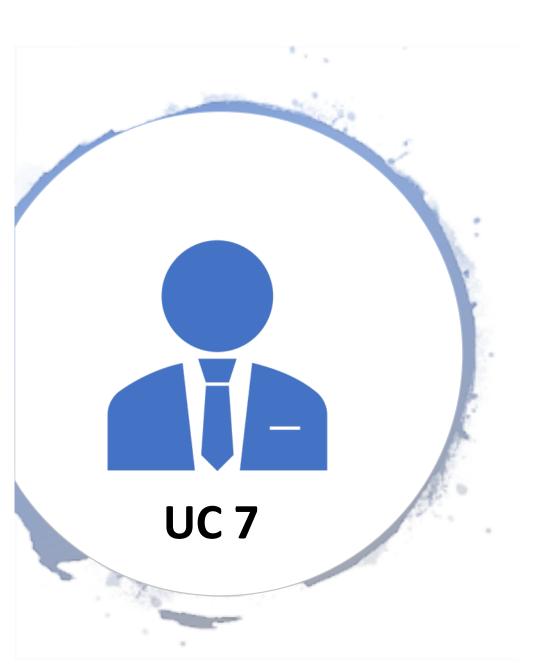
Calculating Wages for a Month

- Assume 20 Working Day per Month

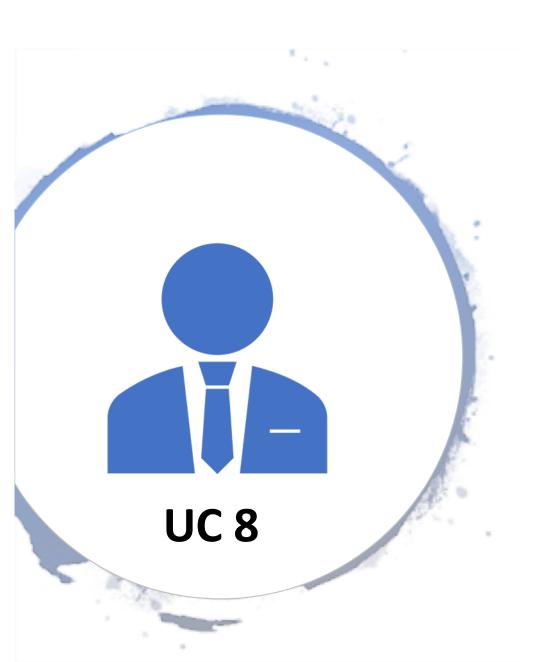


Calculate Wages till a condition of total working hours or days is reached for a month

- Assume 100 hours and 20 days

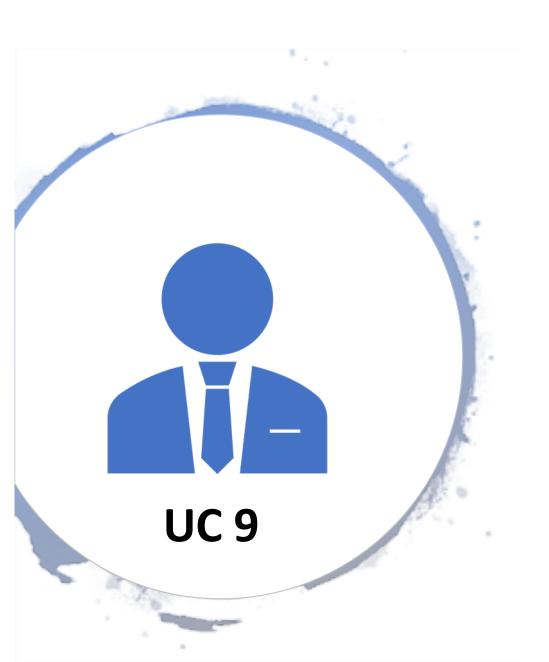


Refactor the Code to write a Class Method to Compute Employee Wage



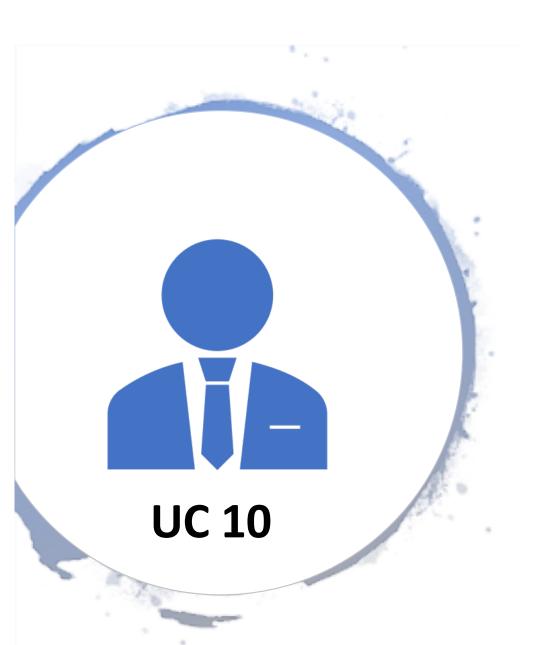
Compute Employee Wage for multiple companies

- Note: Each Company has its own wage, number of working days and working hours per month
- Use Class Method



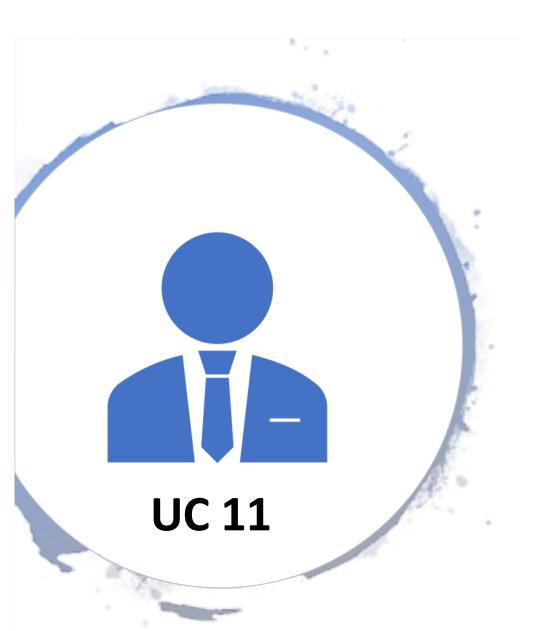
Ability to save the Total Wage for Each Company

Note: You can Create
 EmpWageBuilder for each
 Company



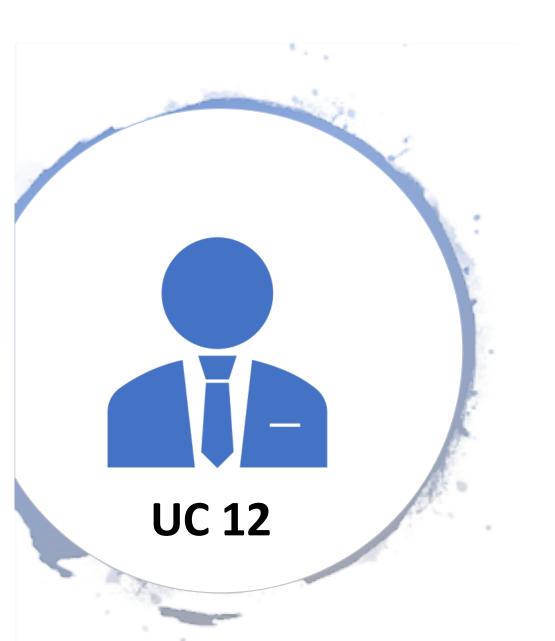
Ability to save the Total Wage for Each Company

Note: You can Create
 EmpWageBuilder for each
 Company



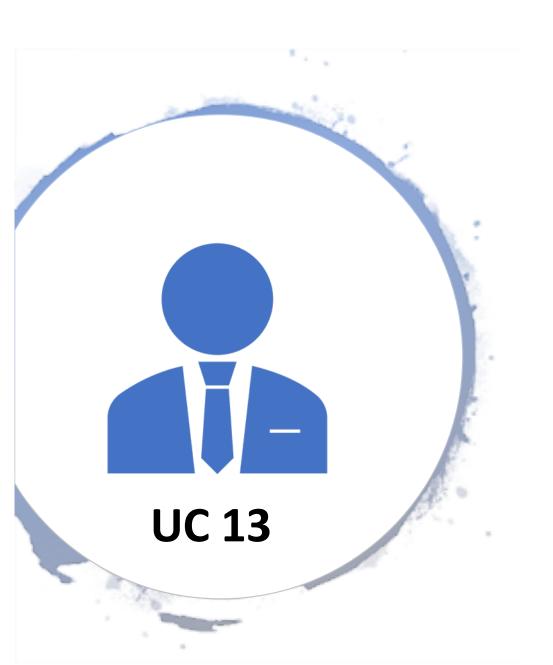
Ability to manage Employee Wage of multiple companies

- Note: Refactor to have one EmpWageBuilder to manage for Wage for multiple Company
- Create CompanyEmpWage class and let EmpWageBuilder has array of many CompanyEmpWage Object

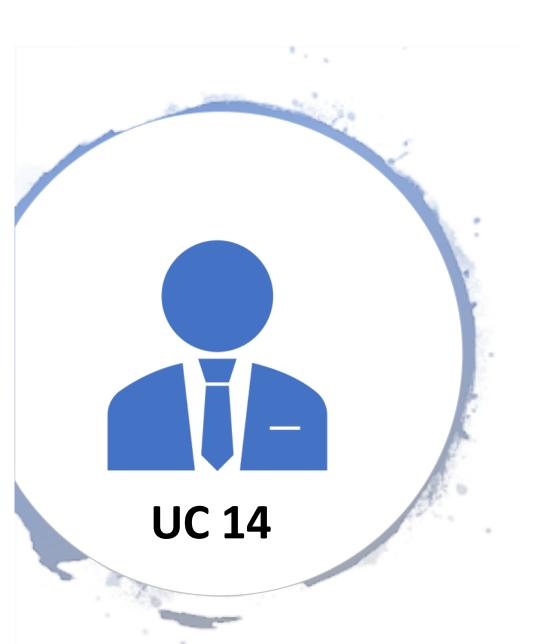


Refactor to have list of multiple companies to manage Employee Wage.

Note: Refactor to use ArrayList instead of array



Store the Daily Wage along with the Total Wage



Ability to get the Total Wage when queried by Company

 Create Method in EmpWageBuilder to get total wage by Company



Thank You