

Task for Exception Handling

Task 1

WAP to validate correct Integer input using Exception Handling

Task 2

Create an `EmployeeException` class whose constructor receives a `String` that consists of an employee's ID and pay rate. Save the file as `EmployeeException.java`. Create an `Employee` class with two fields: `idNum` and `hourlyWage`. The `Employee` constructor requires values for both fields. Upon construction, throw an `EmployeeException` if the `hourlyWage` is less than \$6.00 or over \$50.00. Save the class as `Employee.java`. Write an application that establishes at least three `Employees` with `hourlyWages` that are above, below, and within the allowed range. Display an appropriate message when an `Employee` is successfully created and when one is not.

Task 3

Based on below code implement following functionalities

- 1) User can execute multiple apps
- 2) On the execution of apps, memory should be decreased by minimum memory requirement of that app.
- 3) If user wants to execute another app and sufficient memory not left then it should raise `InsufficientMemoryException` .

```
public class MobilePhone {  
    public static final int MAX_MEMORY_SPACE = 300;  
    public static void main(String[] args) {  
    }  
    public static void appRunner(App app)  
    {  
        System.out.println("Enter the App name which you want to Execute");  
  
        // write code to evaluate whether app can be executed or not  
        // also print the valid msg for non-available app  
    }  
}
```

```
public class App {  
    private String appName;  
    private int minimumMemory; // minimum memory required to execute  
    public App() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
    public App(String appName, int minimumMemory) {  
        super();  
        this.appName = appName;  
        this.minimumMemory = minimumMemory;  
    }  
    public String getAppName() {  
        return appName;  
    }  
    public void setAppName(String appName) {  
        this.appName = appName;  
    }  
    public int getMinimumMemory() {  
        return minimumMemory;  
    }  
    public void setMinimumMemory(int minimumMemory) {  
        this.minimumMemory = minimumMemory;  
    }  
    @Override  
    public String toString() {  
        return "App [appName=" + appName + ", minimumMemory=" + minimumMemory +  
    }  
}
```

```
public class MobilePhoneOS {  
  
    public App[] getAllAppsInfo()  
    {  
        App apps[] = new App[5];  
  
        apps[0] = new App("Whatsapp", 20);  
        apps[0] = new App("Facebook", 520);  
        apps[0] = new App("LinkedIn", 400);  
        apps[0] = new App("Gmail", 120);  
        apps[0] = new App("Camera", 80);  
  
        return apps;  
    }  
}
```