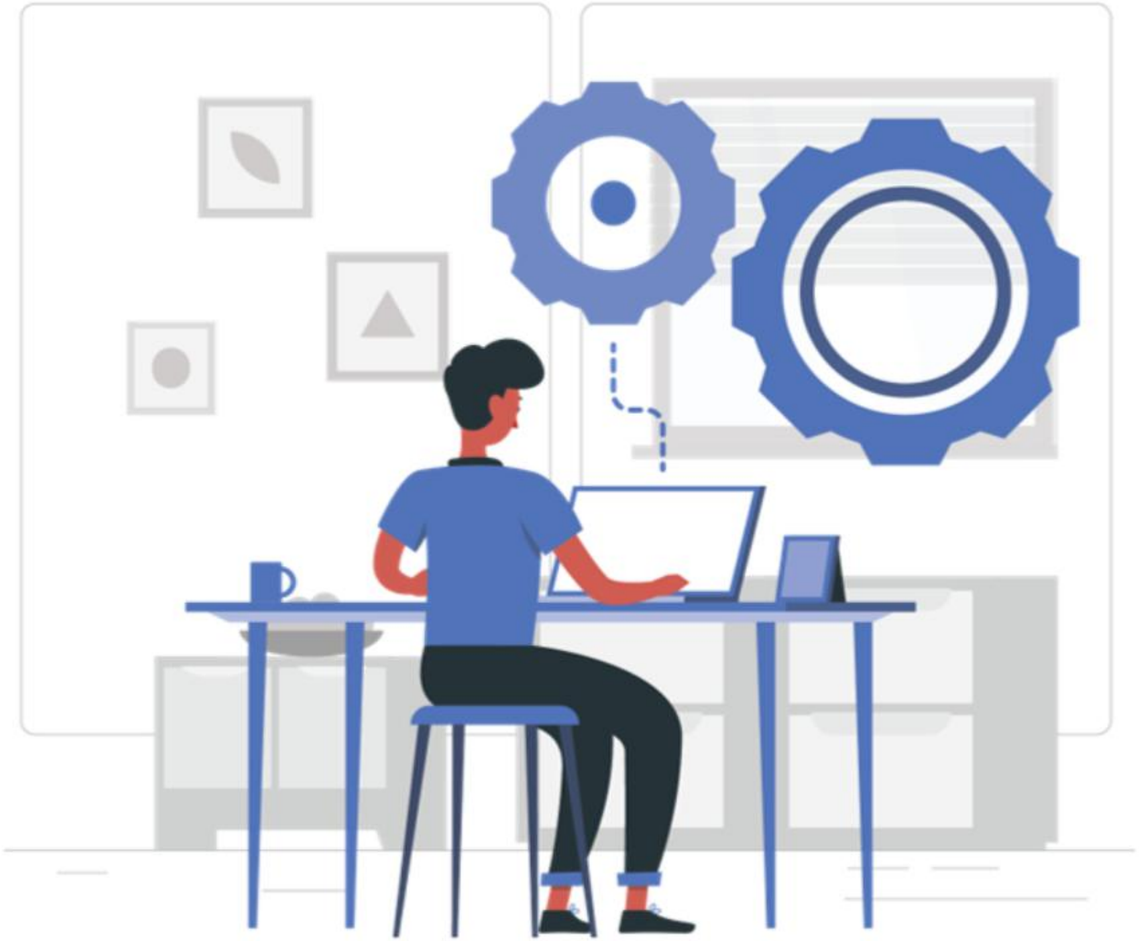
Challenge

Create and Implement a User-Defined Exception



Implementation Environment

- The practice or challenge must be done in the IntelliJ IDE.
- Click here to install IntelliJ.
- You must have access to GitLab.
- Install git to be able to clone and push code to the repository.
- You must be familiar with forking and cloning a git repository.

Implementation Environment

- Cloning a repository from git
  - After forking the boilerplate into your namespace, execute the below command on git bash or command prompt to clone the repository into your local machine.

```
git clone <repository link>
```

- Push the solution back to git
  - After completing the challenge, push the code back to git using the below commands.

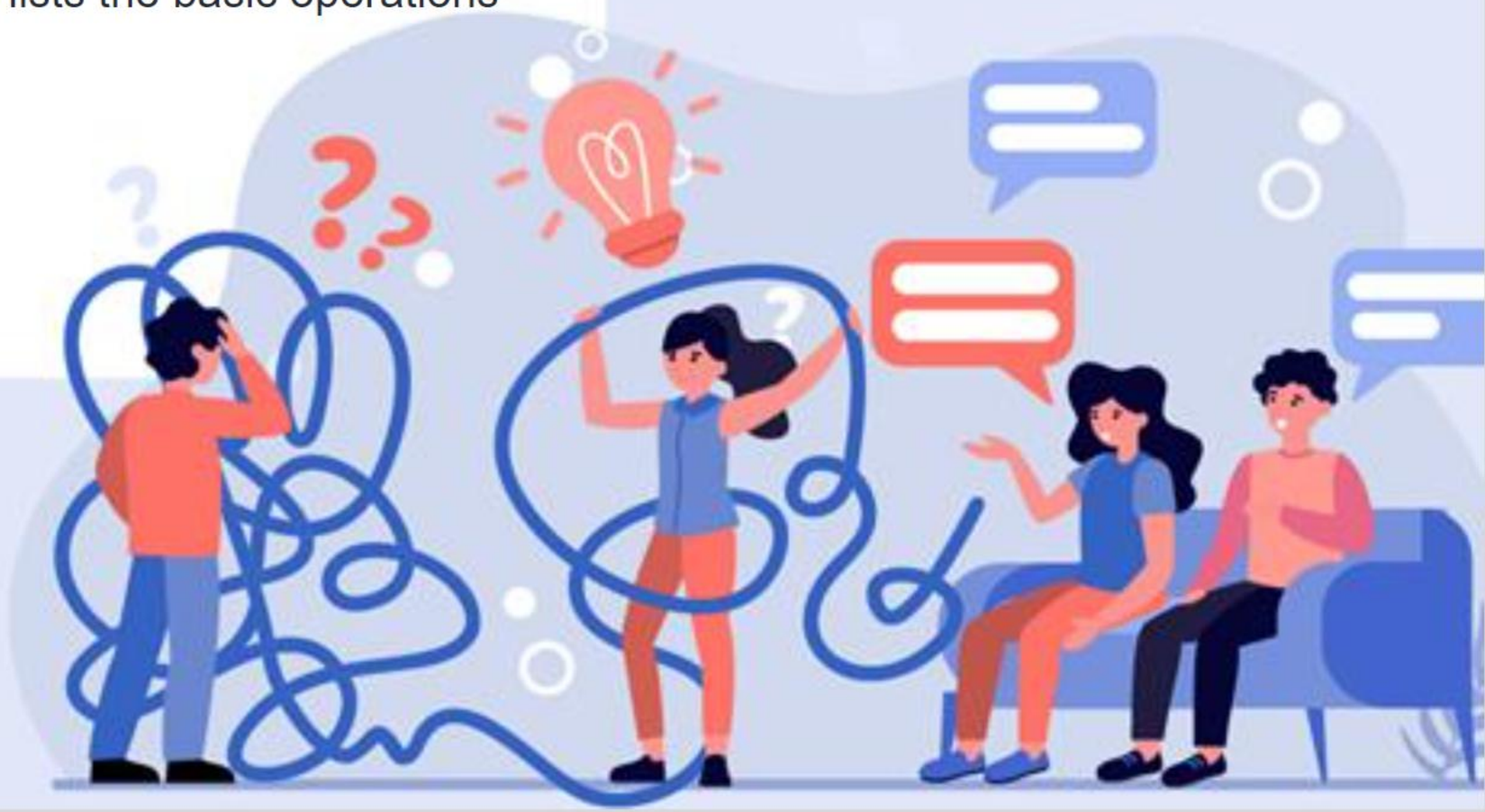
```
git add .
git commit -m "comments on the push"
git push -u origin master
```

Scientific and Mathematic Calculator

John and Bob are playing a game where each has assigned the other the task of creating a calculator.

Design a calculator application that will help perform mathematical and scientific calculations. The next slide lists the basic operations that are performed on the calculator.

CHALLENGE



Instructions for the Challenge

- Click here for the boilerplate.
- Fork the boilerplate using the fork button Fork 282
- Select your namespace to fork the project.
- Clone the project into your local system.
- Open the project in the IntelliJ IDE.
- Work on the solution.
- Execute the test cases given in the test folder.
- Push the solution to git.

Tasks

In the given boilerplate, give implementation, not the specified classes:

- Make the `CalculatorException` class a user-defined - exception class.
- Inside the `MathematicalCalculator` class which will have all mathematical calculations. Handle the appropriate exceptions.
- Write Code inside the `ScientificCalculator` class which will have all the scientific calculations. Handle the appropriate exceptions.

Tasks (Cont'd)

Basic mathematical operations will perform the following tasks:

- Addition of integers and decimal numbers
- Subtraction of integers and decimal numbers
- Multiplication of integers and decimal numbers
- Division of integers (Make sure that a number is not divided by zero) (Handle the situation appropriately)
- Modulo of integers and decimal numbers

Tasks (Cont'd)

- Scientific calculations:
  - Compute the ceil value - If the value entered is 0 or negative, throw the `userdefined` exception.
  - Compute the floor value number - if the value entered is 0 or negative, throw the `userdefined` exception.
  - Square root of a non-negative number - The numbers taken as input must be non-negative. If the number entered is 0 or negative, throw the `userdefined` exception.
  - Compute an exponential expression - The base and exponent must be non-negative.
  - The base and exponent must not be zero.
- Note: Handle all exceptional scenarios.