**IBM HackChallenge –A Mega Hackathon**

**Problem Statement : Method-trace Analyser**  
The challenge is aimed at developing an application to assist developers in debugging code. Method tracing is one of the commonly used post-mortem diagnostic methods to identify problems. It consists of timestamp of entry and exit points for each method invocation. They may also contain stack-trace for each invocation. Depending upon the time for which trace data is collected, the file-size for these traces can be huge. Parsing them manually is a time-consuming and error-prone task.

**Team Size :-**3 Members

**Team Name:-** The Debuggers

**Team Members:-** Shefali Goyal (Team Leader)

Saumya Singh

Shriyansh Sharma

**Project Summary:-**

* For post-mortem debugging, method trace is one of the reliable options. For example, in case of hang / performance related issues - The time taken for each method invocation becomes a key factor. Number of method invocations can be another clue. In case of functional issues – Change in code-flow / Stack-trace. Gathering and comparing these data is time-consuming and prone to human error. The challenge is to create an application to aid developers in debugging Java method trace files.
* We have developed a Java based GUI application which can parse and compare multiple method trace files. The primary goals would be to,
* Compare two trace files: one from failing and passing case each and find out the anomaly.
* Parse one or more trace file and suggest anomalies - I.e. Flag methods which are not completing their execution / Taking longer to execute. This will be helpful in addressing hang and performance related problems.
* Parse one or more trace files and create a tabular and graphical view for the number of times each method is invoked. Comparative view in case of multiple files.
* Compare code-flow and stack trace for failing and passing case and find anomalies.
* **Technical details:**

|  |  |
| --- | --- |
| Language | Java |
| Framework | Java AWT, Java Swing |
| Libraries | Jfreechart, Jcommon, log4j |
| IDE | Eclipse Photon |

* **Proposed outcome/ findings:**

**The** project will display a GUI application for method tracing.

The project will surely be helpful in saving a lot of time in analyzing and debugging a code. It will give us a graphical representation which will be the proof of analyzed code and will show us the performance of the code, the line of code, execution time, errors and exceptions if any.

While execution of the code trace file will be generated itself showing: Date, Time, Line Number, Class/Method Name, the output at each input in code, exceptions and errors if any, total execution time.

**Scope:-**

* This application is used to assist developer in debugging code and reduce the analysis time.
* Method tracing is most commonly used method to identify problems (post mortem debugging).
* It will simplify the complexity of maintain the code.
* It saves investigation effort while debugging with the help of trace files and graphical representation.
* It reduces the human effort while manual debugging as it is automated by this project.
* Its saves time and makes the processing faster.

**Role Of Team Members:**

Shefali Goyal:- Coding

Shriyansh Sharma:- GUI (Swing Portion)

Saumya Singh:- Searching