```
public void run() {
 * Create tidisplay ( finterpreter getHelloPrompt() ); //declared here only to ensure visibility in finally claus
 * this method should be invoked from the * event-di/pass each line of input to finterpreter, and display Introduction to Java
            //fInterpreter's result
private statingutStreamReader inputStreamReader = new InputStreamReader ( SystemCourse Introduction
            BufferedReader stdin = new BufferedReader ( inputStreamReader ) putStream ( buffer );
            boolean hasRequestedQuit = false
            String line snull, lang. Object.
    //Creat List resulte as new ArrayList (); name> //display its data
    frame = try {JFrame("FocusConceptsDemo");
frame.setiwhile (!hasRequestedOnit)" [ava.lang.String">Iterator quarksItr = recoveredQuarks.iterator();
frame.setiwhile (!hasRequestedOnit)" [ava.lang.String">while ( quarksItr.hasNext() ) {
frame.setiwhile (!hasRequestedOnit)" [ava.lang.String">while ( quarksItr.hasNext() ) {
                 line 'stdin readline() dva lang
    //Create and/note that "result" is passed as an "out" parameter
    JComponent hasRequestedQuit = fInterpreter parseInput( line, result );
    newContentPdisplay(Sresult: ) | Sol Maring | les must be | light of lass Console |
    frame.setCorresult clear() on tentPane
    //Display the window.
    frame p catch ( IOException ex ) {
    frame.set System, erraprintin(ex); lass name>java.util.Gregor
               The inheritance tree
                                                                                                    List [aText ] ... (ent.s.
                                                                                           ispl
            finally {class java.util.GregorianCalendar
                                                                                                    = aText.iterator()
                                                                                           extIt
                                                                                                 hasNext() a)ur{uments
public static display(fBYE) ring[ args]
                                                                                                 ter. next() preter) the
    //Scheduleshutdown (stdin event dispatching threa
    //creating and showing this application's GU
    javax. swing. SwingUtilities.invokeLater(new_
                                                                                                                aStdin )
          /// PRIVATE ////
          private static final String fBYE =
          private Interpreter fInterpreter
          /**
  void pri*htDisplay some ttextesto stdouten
     final String[] mvStrings = new String[2]
```

Your Instructor - Rod Davison

50 years experience

Academia (math, linguistics, cognitive science)

Artificial Intelligence R and D

Software Development

Data Analytics – Social Research

Project Manager

Quality and Testing

Business Analysis

Consulting and Training





About the Course

- Introduction to Java Course for IBM developers
 - First of three courses Intermediate and Advanced are next week
 - Covers the basics of Java and OO programming
 - Overview of some related technologies used with Java eg. Spring
 - Content is based on current Java release (Java 20)
- Features from older version are covered as well
 - Differences between current and legacy Java will be discussed
 - Why? Because much of the Java code you might be tasked with supporting or enhancing may have gone into production decades ago
 - You may find yourself supporting code older that you are
- This is a *survey* course
 - Time precludes deep-dives into the various topics
 - The class repo will have reference materials and reading/video lists
- It is assumed that everyone in the class is a programmer
 - Prior experience with Java is not required
 - But a working knowledge of general programming and development tools is assumed



Class Hours

- Review of class hours and breaks
 - Monday to Friday: 9am to 5pm EDT
 - Lunch is fixed at 12-1 EDT to ensure you have a known time to plan calls, meetings, etc
 - Two 15 min breaks mid-morning-ish and mid-afternoon-ish
- Your attendance is being recorded
 - If you are not able to be at class, let the instructor know
 - And let your training coordinator at IBM know as well so you won't be considered AWOL
 - Your attendance is recorded when you log into your LearnQuest student portal



Class Materials

- All of the class materials except for the labs are available at:
 - https://github.com/ExgnosisClasses/IBM-Java-1-Aug14
 - This repo will be available until Sept 18, 2023
- The labs are available at:
 - https://github.com/ExgnosisClasses/Java-1-Labs



Class Protocols

- Learn by doing
 - There will be a strong emphasis on demo and labs
 - How much material we cover is not as important as the amount of material you actually learn
 - Hypothetical problem scenarios will be posed during class for discussion
 - There will be no reading from slides they are only a guide
 - Instead, we will be referring to current online documentation
- Be interactive
 - I will be soliciting your insights, feedback and questions so you have to be present and ready to contribute
 - I like to adapt the course to the needs of students but that requires feedback from you



Lab Environments

- Your LearnQuest Portal contains a link to a VM
 - All the software you need is installed for the class
 - The instructor will use an identical setup to teach from
- If you want to use your own computer, you can
 - The labs assume that you are using Java 20
 - However, time does not allow us to troubleshoot setup issues on your own private computer during class
 - Reach out to your own Tech Support team for help
- The Eclipse IDE is provided in the VM
 - You can customize your VM with another IDE
 - But we cannot troubleshoot any customization issues in class



Introductions

- Please introduce yourself
 - Name you prefer to be known by in class
 - Your area of expertise or specialization (developer, tester, etc.)
 - Programming experience and experience developing in Java and other languages
 - Specific goals or expectations for what you hope to learn





