CPT 237
Advanced Java Programming

Course Description
In this course, we use Java to apply object-oriented programming principles - inheritance, polymorphism, and abstraction, exception handling, JavaFX, and event-driven programming.

Course Focus
Through instruction and application, this course uses Java to provide the student with basic knowledge of object-oriented programming. In this course, students write Java code and use Java classes to implement a variety of algorithms.

Text and References
Introduction to Java Programming, Brief Version, 11th edition
with MyLab Programming and Pearson eText
Liang (author), Pearson (publisher)
ISBN: 9780134694504

Course Outcomes/Learning Outcomes
This course will use various techniques to design programs, which will be written using Java. During the course, the student will learn to do the following:

- Describe objects/classes.
- Use Java classes to model objects.
- Use UML graphical notation to describe classes and objects.
- Define classes and create objects.
- Create objects using constructors.
- Access objects via object reference variables.
- Define a reference variable using a reference type.
- Access an object’s data and methods using the object member access operator (.)
- Define data fields of reference types and assign default values for an object’s data fields.
- Distinguish between object reference variables and primitive data type variables.
- Use the Java library classes Date, Random, and Point2D.
- Distinguish between instance and static variables and methods.
- Define private data fields with appropriate get and set methods.
- Encapsulate data fields to make classes easy to maintain.
• Develop methods with object arguments and differentiate between primitive-type arguments and object-type arguments.
• Store and process objects in arrays.
• Create immutable objects from immutable classes to protect the contents of objects.
• Determine the scope of variables in the context of a class.
• Use the keyword this to refer to the calling object itself.
• Apply class abstraction to develop software.
• Explore the differences between the procedural paradigm and object-oriented paradigm.
• Discover the relationships between classes.
• Design programs using the object-oriented paradigm.
• Create objects for primitive values using the wrapper classes (Byte, Short, Integer, Long, Float, Double, Character, and Boolean).
• Simplify programming using automatic conversion between primitive types and wrapper class types.
• Use the BigInteger and BigDecimal classes for computing very large numbers with arbitrary precisions.
• Use the String class to process immutable strings.
• Use the StringBuilder and StringBuffer classes to process mutable strings.
• Define a subclass from a superclass through inheritance.
• Invoke the superclass’s constructors and methods using the super keyword.
• Override instance methods in the subclass.
• Distinguish differences between overriding and overloading.
• Explore the toString() method in the Object class.
• Discover polymorphism and dynamic binding.
• Describe casting and explain why explicit downcasting is necessary.
• Explore the equals method in the Object class.
• Store, retrieve, and manipulate objects in an ArrayList.
• Implement a Stack class using ArrayList.
• Enable data and methods in a superclass accessible from subclasses using the protected visibility modifier.
• Prevent class extending and method overriding using the final modifier.
• Get an overview of exceptions and exception handling.
• Explore the advantages of using exception handling.
• Distinguish exception types: Error (fatal) vs. Exception (nonfatal) and checked vs. unchecked.
• Declare exceptions in a method header.
• Throw exceptions in a method.
• Write a try-catch block to handle exceptions.
• Explain how an exception is propagated.
• Obtain information from an exception object.
• Develop applications with exception handling.
• Use the finally clause in a try-catch block.
• Use exceptions only for unexpected errors.
• Rethrow exceptions in a catch block.
• Create chained exceptions.
• Define custom exception classes.
• Discover file/directory properties, to delete and rename files/directories, and to create directories using the File class.
• Write data to a file using the PrintWriter class.
• Use try-with-resources to ensure that the resources are closed automatically.
• Read data from a file using the Scanner class.
• Understand how data is read using a Scanner.
• Develop a program that replaces text in a file.
- Read data from the Web.
- Develop a Web crawler.
- Design and use abstract classes.
- Use the abstract Number class to generalize wrapper classes, BigInteger, and BigDecimal.
- Process a calendar using the Calendar and GregorianCalendar classes.
- Specify common behavior for objects using interfaces.
- Define interfaces and define classes that implement interfaces.
- Define a natural order using the Comparable interface.
- Make objects cloneable using the Cloneable interface.
- Explore the similarities and differences among concrete classes, abstract classes, and interfaces.
- Design the Rational class for processing rational numbers.
- Design classes that follow the class-design guidelines.
- Distinguish between JavaFX, Swing, and AWT.
- Write a simple JavaFX program and understand the relationship among stages, scenes, and nodes.
- Create user interfaces using panes, UI controls, and shapes.
- Use binding properties to synchronize property values.
- Use the common properties style and rotate for nodes.
- Create colors using the Color class.
- Create fonts using the Font class.
- Create images using the Image class and to create image views using the ImageView class.
- Layout nodes using Pane, StackPane, FlowPane, GridPane, BorderPane, HBox, and VBox.
- Display text using the Text class and create shapes using Line, Circle, Rectangle, Ellipse, Arc, Polygon, and Polyline.
- Develop the reusable GUI components ClockPane for displaying an analog clock.
- Get a taste of event-driven programming.
- Describe events, event sources, and event classes.
- Define handler classes, register handler objects with the source object, and write the code to handle events.
- Define handler classes using inner classes.
- Define handler classes using anonymous inner classes.
- Simplify event handling using lambda expressions.
- Develop a GUI application for a loan calculator.
- Write programs to deal with MouseEvents.
- Write programs to deal with KeyEvents.
- Create listeners for processing a value change in an observable object.
- Use the Animation, PathTransition, FadeTransition, and Timeline classes to develop animations.
- Develop an animation for simulating a bouncing ball.
- Create graphical user interfaces with various user-interface controls.
- Create a label with text and graphic using the Label class and explore properties in the abstract Labeled class.
- Create a button with text and graphic using the Button class and set a handler using the setOnAction method in the abstract ButtonBase class.
- Create a check box using the CheckBox class.
- Create a radio button using the RadioButton class and group radio buttons using a ToggleGroup.
- Enter data using the TextField class and password using the PasswordField class.
- Enter data in multiple lines using the TextArea class.
- Select a single item using ComboBox.
- Select a single or multiple items using ListView.
- Select a range of values using ScrollBar.
- Select a range of values using Slider and explore differences between ScrollBar and Slider.
- Develop a tic-tac-toe game.
• View and play video and audio using the **Media**, **MediaPlayer**, and **MediaView**.
• Develop a case study for showing the national flag and play anthem.

**Course Goals**

The following list of course goals will be addressed in the course:

• Students will use online exercises to practice using correct Java syntax.
• Students will use Netbeans or a similar IDE to write, debug, compile, and execute Java programs.
• Students will write Java code that includes the following:
  o Classes
  o Objects
  o Packages
  o Inheritance
  o Polymorphism
  o Abstraction
  o Exception handling
  o Event-driven programming
  o JavaFX

**Student Contributions**

Each student will spend 5.5 hours per week in class and possibly another 2 to 3 hours preparing for class and completing assignments to turn in weekly.

Each week students will turn in assignments as specified on Blackboard. Students will also complete assessments in the form of quizzes and exams to demonstrate knowledge of the material.

**Course Evaluation**

The final grade for this course is calculated using the following weights:

• Assignments 40%
• Quizzes 20%
• Exams 20%
• Final Exam 20%

**Course Schedule**

Defines when and how the class meets. Days of the week, time, hybrid, or online.

**Approved by:** Kelli Boniecki _____________________________________________________________________________
Developed/Revised: 08/23/2019
Division Dean for Business & Industrial Divisions

**ADA STATEMENT**

The Technical College of the Lowcountry provides access, equal opportunity and reasonable accommodation in its services, programs, activities, education and employment for individuals with disabilities. To request disability accommodation, contact the counselor for students with disabilities at (843) 525-8228 during the first ten business days of the academic term.

**ACADEMIC MISCONDUCT**

There is no tolerance at TCL for academic dishonesty and misconduct. The College expects all students to conduct themselves with dignity and to maintain high standards of responsible citizenship.

It is the student’s responsibility to address any questions regarding what might constitute academic misconduct to the course instructor for further clarification.

The College adheres to the Student Code for the South Carolina Technical College System. Copies of the Student Code and
Grievance Procedure are provided in the *TCL Student Handbook*, the Division Office, and the Learning Resources Center.

**ATTENDANCE**

1. The College’s statement of policy indicates that students must attend ninety percent of total class hours or they will be in violation of the attendance policy.
2. Students not physically attending class during the first ten calendar days from the start of the semester must be dropped from the class for NOT ATTENDING.
3. Students taking an online/Internet class must sign in and communicate with the instructor within the first ten calendar days from the start of the semester to indicate attendance in the class. Students not attending class during the first ten calendar days from the start of the semester must be dropped from the class for NOT ATTENDING.
4. Reinstatement requires the signature of the division dean.
   a. In the event it becomes necessary for a student to withdraw from the course OR if a student stops attending class, it is the student’s responsibility to initiate and complete the necessary paperwork. Withdrawing from class may have consequences associated with financial aid and time to completion.
   b. When a student exceeds the allowed absences, the student is in violation of the attendance policy. The instructor MUST withdrawal the student with a grade of “W”, “WP”, or “WF” depending on the date the student exceeded the allowed absences and the student’s progress up to the last date of attendance or
   c. Under extenuating circumstances and at the discretion of the faculty member teaching the class, allow the student to continue in the class and make-up the work. This exception must be documented at the time the allowed absences are exceeded.
   d. Absences are counted from the first day of class. There are no "excused" absences. All absences are counted, regardless of the reason for the absence.
   e. A student must take the final exam or be excused from the final exam in order to earn a non-withdrawal grade.

A copy of TCL’s STATEMENT OF POLICY NUMBER: 3-1-307 CLASS ATTENDANCE (WITHDRAWAL) is on file in the Division Office and in the Learning Resources Center.

**ONLINE ATTENDANCE PROCEDURE**

For all online courses, students must complete an assignment designated by the instructor during the first week of classes. The instructor will drop the student from the course if the initial assignment is not completed.

Instructors will withdraw students from the class when 90% attendance is not maintained. Attendance in an online course is defined by regular course access and by timely completion of assignments as required by the instructor. Each student will be expected to access the web class at least once a week and complete 90% of assignments on time. Additional access is encouraged and may be necessary for successful completion of classes.

Failure to log in and complete assignments will result in the student being withdrawn from the course. The instructor will assign a grade of “W,” “WP,” or “WF” based upon the student’s academic standing as of the last date of attendance, which is the last login. **Students are responsible for any financial matters associated with an administrative withdrawal.** If a student fails to email the instructor (using the my.tcl.edu email account) requesting to be dropped from the course and has not submitted the initial assignment required during the first week of class, the instructor will assign a “Never Attended” code in the student information system (Self-Service) no later than ten calendar days after the first day of the class. Students who are dropped as a result of never attending the course are still responsible for all fees associated with the course.

**HAZARDOUS WEATHER**

In case weather conditions are so severe that operation of the College may clearly pose a hardship on students and staff
traveling to the College, notification of closing will be made through the following radio and television stations: WYKZ 98.7, WGCO 98.3, WGZO 103.1, WFXH 106.1, WWVV 106.9, WLOW 107.9, WGZR 104.9, WFXH 1130 AM, WLVR 101.1, WSOK 1230 AM, WAEV 97.3, WTOC TV, WTGS TV, WJWJ TV, and WSAV TV. Students will also receive text alerts and are highly encouraged to verify and/or update contact information at [https://www.tcl.edu/campus-life/campus-security/text-alert/](https://www.tcl.edu/campus-life/campus-security/text-alert/).

**EMERGENCY TEXT ALERT**

Students are automatically opted in to the Emergency Text Message Alert System based on the information on file for you. Students will receive immediate notification of emergency events via text messaging on cell phones. If you would like to verify and/or update your contact information, go to [https://www.tcl.edu/campus-life/campus-security/text-alert/](https://www.tcl.edu/campus-life/campus-security/text-alert/).

**SYLLABUS SAFETY ADDENDUM**

**Purpose**

The purpose of this safety addendum is to provide each student with safety guidelines during an incident, emergency, or disaster at TCL. In addition, it provides students guidelines for lockdown procedures, evacuation procedures, and active shooter.

**Definition**

**An incident** is any event, potential or actual, that may impact normal operations but has no immediate health or life threatening consideration or serious effect on the overall functional capacity of the College. An event of this nature should be reported to the Office of the Vice President for Administrative Services. Also, notify the off-site campus administrator if applicable.

An emergency is any incident, potential or actual, which may endanger life or health or which affects an entire building or buildings, and will disrupt the overall operations of the College. Outside emergency services will probably be required, as well as major efforts from campus support services. Major policy considerations and decisions will usually be required from the college administration during times of crises. An emergency should be reported immediately by directly using 911 if life or health/injury considerations exist and then to the Office of the President or Vice President for Administrative Services as quickly as possible. Also, notify the off-site campus administrator if applicable.

A **disaster** is any event or occurrence that has taken place and has seriously impaired or halted the operations of the College. In some cases, mass personnel casualties and severe property damage may be sustained. A coordinated effort of all campus-wide resources is required to effectively control the situation. Outside emergency services will be essential. In all cases of disaster, an Emergency Control Center will be activated, and the appropriate support and operational plans will be executed. The disaster should be immediately reported, first by calling 911 and then to the Office of the President or Vice President for Administrative Services. Also, notify the off-site campus administrator if applicable.

**Types of Emergencies**

- Hurricane
- Tornado
- Fire
- Biochemical or Radiation Spill
- Explosion/Bomb
- Downed Aircraft (crash which directly impacts campus operations)
- Utility Failures
- Violent or criminal behavior
- Psychological Crisis

**Procedures**

**Building Evacuation**

1. Building evacuations occur when an alarm sounds and/or upon notification by Security or the Emergency Director.

2. When the building evacuation alarm is activated during an emergency, individuals should exit according to the building evacuation plan and alert others to do the same.

3. Once outside, individuals should proceed to a clear area that is at least 500 feet away from the affected building. Streets, fire lanes, hydrant areas and walkways should be kept clear for emergency vehicles and personnel.

4. Individuals should not return to an evacuated building unless told to do so by Security or the Emergency Director.

5. Individuals should assist persons with disabilities in exiting the building. Elevators are reserved for disabled persons

**Campus Evacuation**

1. A uniformed Security Guard, the Emergency Director, or an Emergency Resource Team member will announce evacuation of all or part of the campus grounds.

2. All persons (students and staff) are to immediately vacate the campus, or in the case of a partial evacuation relocate to another part of the campus grounds as directed.

**Lockdown**

1. Clear the halls
2. Report to the nearest classroom/office
3. Assist those needing special assistance
4. Ensure classroom/office doors are closed and locked
5. Turn off lights
6. Stay away from doors and windows (out of the line of sight)
7. BE QUIET and follow instructor’s directions
8. Silence cell phones
9. Wait for the “All Clear” before leaving