

Ministry of Higher Education
Pkfokam Institute of Excellence
Department of Computing and Software Engineering
Course Code: CS 1302
Course Name: Programming and Problem Solving II
Semester: Fall 2021



SYLLABUS

Instructors Information:

1. *Instructor's Name:* Mr. FOGAING Christian
Email: christian.fogaing@pkfinstitute.com
Office Hours: By appointment
2. *Instructor's Name:* Mr. MEKONTSO Herman
Email: metchiha@gmail.com
Phone: (+237)693 031 561
Office Hours: By appointment

Lecture Meeting Times:

Normal class: Saturday 11:00 am – 01:00pm

Required course material:

Required Text: Introduction to Java Programming, Comprehensive Version, 10th Edition By Y. Daniel Liang, Pearson Publishing, 2015, ISBN#: 978-0-13-376131-3

Pre - requisite: CS1301

Overview:

The second course in computer science provides coverage of more advanced topics of object-oriented programming. This includes inheritance and polymorphism, files and exception handling, and parameterized types (generic programming). Elementary data structures (linked lists, stacks, and queues) are introduced to solve application problems. Graphical user interfaces and event driven programming are also introduced. Students must continue to use good programming style including proper documentation. Java runtime environment (JRE) and libraries security issues are briefed.

Learning Outcomes:

1. Demonstrate more advanced skills in programming, including the ability to handle multi-dimensional arrays.
2. Design and use classes, including inherited classes.
3. Demonstrate file handling and exception coding in a program.
4. Recognize the difference between iterative and recursive methods and use them correctly in a program
5. Demonstrate the ability to use linked lists, stacks, and queues in problem solutions.

6. Compile and run Java programs in a Windows environment.
7. Understand both the "program driven" and "event-driven" approaches to user interaction, including the relationship between event-driven programs and Graphical User Interfaces
8. Realize vulnerabilities in libraries or JRE when programming.

Grading Plan:	1000 points		Grading Scale:	These are minimums. The final grades will be determined by distribution.
150 points	Assignments		900 +	A
150 points	Quizzes		800 - 899	B
150 points	Test 1		700 - 799	C
150 points	Test II		600 - 699	D
300 points	Final		below 600	F
100 points	Class Participation			

Schedule and Topic Coverage:

Week	outlines	Reference in the text book
1	Inheritance and Polymorphism	Chapter 11
2	Exception handling and Text IO	Chapter 12
3	Abstract Classes and Interfaces	Chap 13
4	File I/O, with an emphasis on Binary files	Chap 17
5	Recursion	Chapter 18
6	Revisions and Test I	
7	JavaFX Basics	Chap 14
8	Event Driven Programming & Animations	Chap 15
9	JavaFX UI Controls	Chap 16
10	Java FX Revisions	
11	Introduction to Scene Builder	Ref Book on Scene Builder
12	Revisions and Test II	
13	Generics	Chap 19
14	Lists, Stacks, Queues & Priority Queues	Chap 20 (Part I)
15	Revisions and Quiz	

Tentative Class Calendar

Wk	Wk-D1	Wk-D2
1	Inheritance and Polymorphism	Inheritance and Polymorphism Cont.
2	Exceptions and Text IO Files	Exceptions and Text IO Files Cont.
3	Abstract Classes and Interfaces	Abstract Classes and Interfaces Cont
4	Binary IO Files	Quiz I
5	Binary IO cont.	Recursion
6	Recursion Cont	Test 1

7	JavaFX Basics	JavaFX Basics – cont
8	Event Driven Programming & Animations	Event Driven Programming & Animations (Cont.)
9	Revisions	JavaFX UI Controls
10	JavaFX UI Controls - Cont	QUIZ II
11	Introduction to Scene Builder	Introduction to Scene Builder cont
12	Test 2	Generics
13	Generics Cont.	Lists, Stacks, Queues & Priority Queues
14	Lists, Stacks, Queues & Priority Queues Cont	Quiz III
15	REVISIONS	REVISIONS

Class Rules:

1. Students are responsible for all announcements and assignments made in class.
2. Students are expected to do their own work for all assignments unless otherwise indicated by instructor. Group discussion and study of the assignments are permitted, but, when you begin to prepare your assignment to turn in, all collaboration must cease. If collaboration is suspected, the grade will be a 0. Multiple occasions of collaboration will earn you an academic dishonesty F for the course.
3. All assignments are due *AT MIDNIGHT* on the day indicated.
4. **Late programs will be accepted UP TO ONE WEEK AFTER THE ORIGINAL DUE DATE.**
5. Late programs will be penalized 5 points per day (including weekends, not including holidays), up to the final acceptance date. After that, the value of the program will be 0 point.
6. Early programs are rewarded 5 points per day (including weekends, not including holidays), up to 3 days early. Should you turn in an assignment a week early, you will still only earn 15 additional points. This is to encourage you to START EARLY and DO NOT PROCRASTINATE! Programs must compile and produce correct results for a majority of the test cases to get early bonus points.
7. If a class is cancelled for any reason, any assignment due that day will immediately be due the next scheduled class period. Any topic or test scheduled for that day will occur the next scheduled class period.
8. Class attendance and participation is expected. Meaningful and relevant class discussion is strongly encouraged as can be seen in the grading plan.
9. In case of justifiable unforeseen, student should report their absence to the class delegate or directly to the teacher, specifying the reason why they will be absent. Students are expected to do that before that day, otherwise, they will irreversibly be marked absents.
10. There is no reason for not doing an assignment. Students are expected to do their assignments even if they were in class or were sick the day of the course.
11. All current PKFOKAM INSTITUTE OF EXCELLENCE policies will apply.

Assignment Policies:

When turning in your assignments, **it must start with a cover sheet** followed by the program listing (source code with comments), followed by the output. All assignments must be stapled, in a binder or otherwise fastened together. Program assignments will be graded heavily for correct results, but emphasis will also be placed upon accurate and neat documentation as well as effective and proper use of the Java language.

All programming assignments must include the student's name and the assignment number.

For Lab Assignments, please add the following line to the end of your source program:

System.out.println("Coded by _____"); //enter your(s) name(s) here

Remember that everyone is working on the same lab...without your name, We don't know whose it is! We need the front page from your lab assignment, followed by what is required for each lab.

Turning in Programs:

1. Turn in programs on the day of class, at the beginning of class. (preferred)
2. Turn in programs to the PKFOKAM front office, and ask that they be time/date stamped.
3. You may email your assignments to the lab instructor at metchiha@gmail.com (source code and output). A printed copy **must** be turned in to me at the next class period or it will be counted late as when the printed copy is received. If the printed copy does not match the email, the printed copy will be graded and counted as received on that date.
4. You will be required to submit a hardcopy of the assignment. In addition, all students must submit an executable version of an assignment upon request.

Assignment Cover Sheet

When turning in your programming assignments (not labs), you must include a signed copy of the cover sheet. Basically in this course, the honor code pledge asserts that work which you submit as being your own really is your own. Use the lab time to learn the material and gain the confidence to complete the programming assignments. The programming labs must be designed and coded by the individual student – if you need help, please see the instructor or lab assistant. You are not allowed to copy another student's work or have someone else do the assignment for you. Tests/quizzes must be taken by each individual student in a supervised classroom with no electronic devices available.

In the case of programs there is sometimes a gray area as to what constitutes "your own work." Clearly, taking someone else's code and permuting it by changing comments, procedure order, variable names, etc. is rampant plagiarism; it is not "your own work." Suppose instead that you talk with a friend about the algorithm, and then code up that algorithm, is that plagiarism? I would say no. The intellectual challenge of this course is to take abstract algorithms, objects, or ideas and make them work. You may freely get help understanding the algorithm, objects, or idea - but you must make it work. The line I draw is: "if you actually look at another person's code (or let someone look at yours) you are stepping over the line."

If you are found in violation of the honesty pledge, you will receive an academic dishonesty "F" for the course.



Name: _____

CS1302 – Programming and Problem Solving II

Assignment #_____

Instructors: M. FOGAING / M. MEKONTSO

Honor Pledge

On my honor as a student, I have neither *given* nor *received* unauthorized aid on this assignment.

Signed _____