Syllabus

Catalog Description:
Introduces basic procedural and object-oriented programming. Topics include: variables, data types, assignments, conditionals, loops, arrays, input/output, static methods, libraries, recursion, data types, API, classes, access modifiers, instance variables, constructors, instance methods, testing, encapsulations, immutability, interface inheritance, implementation inheritance, exceptions, assertions, analysis of algorithms, order of growth, memory usage, binary search, insertion sort, merge sort, stacks, array implementation of stacks, linked list implementation of stacks, queues, generics, autoboxing, iteration, symbol tables, hash tables, binary search trees, examples and applications.

Expected outcomes:
- Students understand and apply basic procedural programming principles.
- Students understand and apply basic object-oriented programming principles.
- Students understand, analyze, compare and apply basic data structures and their implementation.
- Students understand, analyze, compare and apply basic sorting methods.
- Students understand, analyze, compare and apply basic searching methods.

Prerequisites:
The course requires basic information technology knowledge. This is a remedial programming course.

Sections
AIT 502-002: (Flipped classroom with active learning)

Course Coordinator/Instructor
Dr. Mihai Boicu, Ph.D. - Phone: 703 993 1591 Email: mboicu@gmu.edu

Textbook
Introduction to Programming in Java: An Interdisciplinary Approach 1st Edition
by Robert Sedgewick (Author), Kevin Wayne (Author)

Amazon: https://www.amazon.com/gp/product/0321498054/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&tag=introcs-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0321498054

Course web site: https://introcs.cs.princeton.edu/java/home/
Additional readings, tutorials and online materials will be recommended during the course.

### Schedule

<table>
<thead>
<tr>
<th>Topic</th>
<th>Week / Description</th>
<th>Readings</th>
</tr>
</thead>
</table>
| Fundamentals               | 1. Data types, Conditional and Loops | Chapter 1. Elements of programming  
1.1. Elements of programming  
1.2. Built-in types of data  
1.3. Conditionals and Loops |
|                            | 2. Arrays                     | 1.4. Arrays                                                              |
|                            | 3. Input and output            | 1.5. Input and Output                                                   |
|                            | 4. Case study and review       | Chapter 1. Elements of programming  
1.6. Random Web Surfer       |
| Exam 1                     | 5. Mini Exam 1                 | Short one hour exam covering Chapter 1                                   |
| Functions                  | 5. Methods                    | Chapter 2. Functions  
2.1. Static methods                                                     |
|                            | 6. Libraries                  | 2.2 Libraries and Clients                                               |
|                            | 7. Recursion                  | 2.3. Recursion                                                          |
|                            | 8. Case study                 | 2.4. Case study: Percolation                                            |
| Exam 2                     | 9. Mini Exam 2                 | Short one hour exam covering Chapters 1-2                               |
3.1. Using data types                                                   |
|                            | 10. Creating classes          | 3.2. Creating data types                                                |
|                            | 11. Designing data types      | 3.3. Designing data types                                               |
|                            | 12. Case study                | 3.4. N-body simulation                                                  |
| Exam 3                     | 13. Mini Exam 3               | Short one hour exam covering Chapters 1-3                               |
| Data Structures            | 13. Performance analysis and sorting and searching | Chapter 4. Algorithms and data structures  
4.1. Performance  
4.2. Sorting and searching                                           |
|                            | 14. Stacks, queues, symbol tables | 4.3. Stacks and queues  
4.4. Symbol tables                                                       |
| Final exam                 | Cumulative exam               | Full time exam covering Chapters 1-4                                    |
Note: The schedule may be changed during the semester to accommodate specific class needs. All changes will be posted on the Blackboard and communicated by email.

Grading:
The students will be assigned a grade based on a weekly portfolio:

- Weekly Assignments (60% required + 10% bonus points = 5% * 14)
- Class/Online participation (5% required + 9% bonus points = 1% * 14)
- Short Exams (15% = 3 x 5%)
- Final exam (20%)
- Bonus points might be provided during class for participation, extra assignments, extracurriculum activities, enrichment activities at the discretion of the instructor

The grading scale for this course is:

<table>
<thead>
<tr>
<th>Numeric Grade</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>97 – 100%</td>
<td>A+</td>
</tr>
<tr>
<td>93 – 96%</td>
<td>A</td>
</tr>
<tr>
<td>90 – 92%</td>
<td>A-</td>
</tr>
<tr>
<td>87 – 89%</td>
<td>B+</td>
</tr>
<tr>
<td>83 – 86%</td>
<td>B</td>
</tr>
<tr>
<td>80 – 82%</td>
<td>B-</td>
</tr>
<tr>
<td>77 – 79%</td>
<td>C+</td>
</tr>
<tr>
<td>73 – 76%</td>
<td>C</td>
</tr>
<tr>
<td>70 – 72%</td>
<td>C-</td>
</tr>
<tr>
<td>60 – 69%</td>
<td>D</td>
</tr>
<tr>
<td>0 – 59%</td>
<td>F</td>
</tr>
</tbody>
</table>

Hardware and Software requirements

For all sections you must have a personal computer with internet connection. It is strongly recommended that you have a powerful enough laptop on which you can perform code development that you can bring to class.

For the online section you must have a personal computer with internet connection, with speakers and microphone.

We require either a Mac OS X or Windows 10 computer.
**Course Delivery Methods**

The course will be delivered using various methods. You must have your MASON email account activated and you must check your email daily for announcements related to the course. You must have access to Blackboard Learning System and to know how to use its features.

There are video presentations posted on the Blackboard. You must have an environment in which you can watch these videos.

You will have several assignments and assessments to be performed each week. A summary of weekly requirements will be sent at the beginning of the week.

**COURSE CANCELED (SNOW DAYS)**

If the courses are canceled the first option is to have a synchronous meeting online during the same times. If you cannot be online the course will be recorded and posted on the course Blackboard site.

**Exams**

There are two main exams (midterm and final). The exams must be taken in class at the schedule date. For online sections the exam may be taken also in a pre-approved testing center. Exceptions must be well documented and approved based on MASON exams guidelines.

**Intellectual Property**

There is a strong recommendation that all work in the class projects to be done based on an open source license (e.g. Academic Free License [http://en.wikipedia.org/wiki/Academic_Free_License](http://en.wikipedia.org/wiki/Academic_Free_License)). This will allow a rich, shared exchange of ideas and will allow each member of the class to further benefit with no restriction from the work performed in the class.

**Privacy**

Instructors respect and protect the privacy of information related to individual students. Specific issues relating to an individual student will be discussed via email, telephone or in person. Instructors will not discuss issues relating to an individual student with other students (or anyone without a need to know) without prior permission of the student. There is no guarantee related to the security of email and telephone conversations.

Assessable work other than final exams will be returned to individual students directly by the Instructor (or by a faculty or staff member or a Teaching Assistant designated by the Instructor, or via another secure method). Under no circumstances will a student’s graded work be returned to another student.

Faculty and staff will take care to protect the privacy of each student’s scores and grades.
Because of the nature of this class, some work performed by the student will be published and discussed in the class. Other students will be able to make comments and suggestions related to the published work, without seeing the actual grade the student earned for the work.

**Disability Accommodations**

The Office of Disability Services (ODS) works with disabled students to arrange for appropriate accommodations to ensure equal access to university services. Any student with a disability of any kind is strongly encouraged to register with ODS as soon as possible and take advantage of the services offered.

Accommodations for disabled students must be made in advance – ODS cannot assist students retroactively, and at least one week's notice is required for special accommodations related to exams. Any student who needs accommodation should contact the Instructor during the first week of the semester so the sufficient time is allowed to make arrangements.

**Honor Code**

All members of the Mason community are expected to uphold the principles of scholarly ethics. Similarly, graduating students are bound by the ethical requirements of the professional communities they join. The ethics requirements for some of the communities relevant to Applied IT graduates are available via the following links:

- ACM Code of Ethics and Professional Conduct
- IEEE Code of Ethics
- EC-Council Code of Ethics

On admission to Mason, students agree to comply with the requirements of the Mason Honor System and Code. The Honor Code will be strictly enforced in this course. Honor Code cases are heard by a panel consisting of students – students who meet the requirements are encouraged to nominate themselves to serve on the Honor Committee.

Any use of the words or ideas of another person(s), without explicit attribution that clearly identifies the material used and its source in an appropriate manner, is plagiarism and will not be tolerated. There is a "zero tolerance" policy for plagiarism within The Volgenau School. The Instructor reserves the right to use manual and/or automated means (including such services as Turnitin.com) to detect plagiarism in any work submitted by students for this course, and to direct Teaching Assistants and/or other faculty and/or staff members to do likewise in support of this course.

For this course, the following requirements are specified:

- All assessable work is to be prepared by the individual student, unless the Instructor explicitly directs otherwise.

---

1 Available at [http://catalog.gmu.edu/](http://catalog.gmu.edu/) and related Mason Web pages.
AI T502 – Programming Essentials

Syllabus

- All work must be newly created by the individual student for this course for this semester. Any usage of work developed for another course, or for this course in a prior semester, is strictly prohibited without prior approval from the instructor.

- For team work a summary at the end of the submission must identify mutually agreed individual contributions.

Students may seek assistance with assigned work (and are encouraged to do so if they feel the need), provided:

- The directions for the assigned work do not prohibit such assistance.

- Such assistance is acknowledged in the submitted work, clearly identifying the person(s) giving assistance and the nature of the assistance given.

- Any work to be submitted is prepared entirely an exclusively by the student submitting it. Students are expressly prohibited from sharing any assessable work for this course in any manner with other students (except students assigned as Teaching Assistants or Undergraduate Peer Mentors to this course and the student’s section), unless all students involved have had their work graded and returned by the Instructor, or the Instructor has explicitly approved such sharing.