CS 258 – Introduction to Object Oriented Programming
Syllabus for Spring Semester, 2017

Syllabus Change Policy: This syllabus may be subject to change with reasonable advanced notice.

Course Description: Introduces the fundamental concepts of programming from an object-oriented perspective. Through the study of object design, this course introduces data abstraction, inheritance, overriding, programming flow of control, operator precedence, and simple data structures such as lists and arrays. This course also introduces the basics of human-computer interfaces, graphics, and the social implications of computing, along with significant coverage of software engineering. 3 Hours Credit.

Prerequisites: None

USI Computer Science Program Goals:
This course supports the expected characteristics, capabilities and skills for computer science graduates in the USI Computer Science program of study in the following ways:

1.) Mastery of CS technical foundations.
Students will continue to build knowledge and understanding of computer science areas including algorithms and design, fundamental programming concepts, fundamental data structures, software development methods, and ethics.

2.) Recognition of common CS themes and principles.
Students will continue to be able to recognize and identify recurring themes (such as abstraction, complexity, and recursion) that present themselves in many diverse computer science domains, recognize and identify recurring principles (such as sharing a common resource, security, and concurrency) that present themselves in many diverse computer science domains, and apply recurring themes and principles where appropriate.

3.) Recognition of interplay between theory and practice.
Students will continue to be able to explain and illustrate how theory and practice influence each other.

4.) Effective problem solving and critical thinking skills.
Students will continue to be able to identify and use relevant concepts, identify and use relevant information, select and use appropriate actions or operations, interpret clearly and logically from prior activities, develop qualitative and quantitative assessment and
develop solutions to problems not only in single levels of details, but also multiple levels of detail and abstraction.

5.) Commitment to life-long learning, and professional and ethical responsibility. Students will continue to be introduced to the concept that they must develop a plan that establishes how they will maintain skills to remain relevant in the CS field, be able to identify cross disciplinary opportunities between CS and other fields, and to recognize, analyze, and address the intellectual, professional, economic, social, legal, ethical, and cultural issues associated with computing.

6.) Effective communication and organizational skills. Students will continue to be able to make presentations, whether oral, written or electronic communications that are organized, relevant, and customized for the audience using appropriate CS terminology.

Course Learning Objectives:
This course, in particular, contributes to the Computer Science Program Learning Objectives in the following ways:

- Discuss items pertaining to the history of computer science, programming languages, and the compilation process,
- Apply the C# object-oriented programming language,
- Determine the components of the object-oriented paradigm, including objects, classes, methods, abstraction, encapsulation, subclassing, inheritance, polymorphism and parameter passing,
- Implement fundamental data structures, including primitive types, arrays, strings and records,
- Implement expressions and assignments,
- Implement the fundamental programming constructs, including basic syntax and semantics of C#, variables, types, expressions, assignments, simple input and output, conditional and iterative control structures and structured decomposition,
- Implement algorithms and problem-solving,
- Use APIs (class libraries and packages for graphics and GUI applications) and software engineering tools, such as IDEs,
- Use step-wise refinement for multiple-step implementation of algorithms,
- Understand the ethics and responsibilities of computer professionals,
- Increase critical thinking skills, analytical problem-solving skills and awareness of computer-related ethics.

Instructional Learning Objectives:
This course, in particular, includes instruction in the following knowledge units identified in the ACM Computer Science CS2013 Knowledge Units:
- Object-oriented programming, basic type systems, algorithms and design, fundamental programming concepts, fundamental data structures, development methods, professional ethics, and history.

**Course Materials:**

**Evaluation:**
Performance in this course will be evaluated on the following basis:

<table>
<thead>
<tr>
<th>Weight</th>
<th>Exams</th>
<th>45%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Assignments/Quizzes</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>Attendance</td>
<td>5%</td>
</tr>
</tbody>
</table>

Grades will be assigned as follows:
90-100% A, 80-89% B, 70-79% C, 60-69% D, <60% F

Assignments will be assessed using assessment score sheets available on-line under Blackboard Course Documents. Exams, which may include matching and multiple-choice questions, will be assessed by correct responses. Exposure to Instructional Learning Objectives (topic areas) will be assessed by monitoring class attendance.

**Tentative Schedule:** (See Blackboard for detailed dates)
Computer Science Program
Common Syllabus Statements

Computer Science Program Statements

In-class conduct
Disruptive and/or distractive behavior will not be tolerated in this class.

Mobile devices policy
All mobile devices must be turned off during this class.

Communications
Blackboard will be used as the repository for all materials necessary for successful completion of this course. Please refer to Blackboard for any schedule, assignment, exam and/or additional contact information.

Academic Integrity
Academic dishonesty (plagiarism, cheating, copying, etc.) will not be tolerated in this class under any circumstances. Any student found violating this rule will receive a failing grade for the semester and will be expelled from this class. You are expected to do your own work on assignments, tests and programs. Helping others or receiving help from others when debugging programs is not dishonest; however, writing-out corrections and/or copying software for others is dishonest. Please also refer to: www.usi.edu/deanofstudents/academic-integrity]

University Statements

Disability Accommodations
If you have a disability for which you may require academic accommodations for this class, please register with Disability Resources (DR) as soon as possible. Students who have an accommodation letter from DR are encouraged to meet privately with course faculty to discuss the provisions of those accommodations as early in the semester as possible. To qualify for accommodation assistance, students must first register to use the disability resources in DR, Science Center Rm. 2206, 812-464-1961, www.usi.edu/disabilities. To help ensure that accommodations will be available when needed, students are encouraged to meet with course faculty at least 7 days prior to the actual need for the accommodation. However, if you will be in an internship, field, clinical, student teaching, or other off-campus setting this semester please note that approved academic accommodations may not apply. Please contact Disability Resources as soon as possible to discuss accommodations needed for access while in this setting.
Title IX – Sexual Misconduct
USI does not tolerate acts of sexual misconduct, including sexual harassment and all forms of sexual violence. If you have experienced sexual misconduct, or know someone who has, the University can help. It is important to know that federal regulations and University policy requires faculty to promptly report incidences of potential sexual misconduct known to them to the Title IX Coordinator to ensure that appropriate measures are taken and resources are made available. The University will work with you to protect your privacy by sharing information with only those who need to know to ensure we can respond and assist. If you are seeking help and would like to speak to someone confidentially, you can make an appointment with a counselor in the University Counseling Center. Find more information about sexual violence, including campus and community resources at www.usi.edu/stopsexualassault.
GENERAL INFORMATION FOR ALL CLASSES  
Spring Semester, 2017

INSTRUCTOR:  Srishti Srivastava, Assistant Professor of Computer Science  
OFFICE:  BE 2085  
OFFICE PHONE:  812-228-5032  
EMAIL:  fsrishti@usi.edu  
OFFICE HOURS:  T TH 4:15 – 5:30pm  
Or other times by appointment

Class Participation: Your attendance is required for all scheduled class meetings. The whole class benefits from your attendance and participation. Attendance will be taken every scheduled class meeting and your attendance and class participation will affect your final grade. I will regularly introduce material not found in your text and/or class materials and expect you to be familiar with it. I will also regularly give unscheduled quizzes on class materials.

Missed Classes: If you miss a scheduled class meeting, it is your responsibility to find out the material and assignments that were covered in your absence.

Late Assignments: Assignments are due at the beginning of class on the day due. Any assignment submitted after the beginning of class will be considered late and will be accepted only with a penalty of 10% for each class meeting day (or partial day) late.

Missed Exams: If you miss a scheduled exam, you will need a doctor’s or similar excuse to take the exam at a later date or you will receive a 0% for that exam. Please notify me, in advance, if you will not be able to attend class on the day of a scheduled exam so that we may arrange your make-up exam date.

Office Hours: I am available during posted office hours for any questions you may have. I am also available for appointments, scheduled in advance, at our mutual convenience. I will always be available to help you perform to the best of your ability in this class.
USI Computer Science Program Goals 2017
Characteristics of USI Graduates

At a broad level, the expected characteristics of USI computer science graduates include the following:

1. **Mastery of computer science technical foundations.**
   Graduates have a mastery of computer science as described by the core of the Body of Knowledge in ACM Computer Science Curriculum 2013. [1]

2. **Recognition of common computer science themes and principles.**
   Graduates recognize a number of recurring themes, such as abstraction, complexity, and evolutionary change, and a set of general principles, such as sharing a common resource, security, and concurrency. Graduates understand that these themes and principles have broad application to the field of computer science and should not consider them as relevant only to the domains in which they were introduced.

3. **Recognition of interplay between theory and practice.**
   Graduates understand the interplay between theory and practice and the essential links between them. Graduates understand how theory and practice influence each other.

4. **Capability of assessment from a system-level perspective.**
   Graduates think at multiple levels of detail and abstraction. This understanding transcends the implementation details of the various components to encompass an appreciation for the structure of computer systems and the processes involved in their construction and analysis. They recognize the context in which a computer system may function, including its interactions with people and the physical world.

5. **Effective problem solving and critical thinking skills.**
   Graduates understand how to apply the knowledge they have gained to solve real problems, not just write code and move bits. They are able to design and improve a system based on a quantitative and qualitative assessment of its functionality, usability and performance. They realize that there are multiple solutions to a given problem and that selecting among them is not a purely technical activity, as these solutions will have a real impact on people’s lives. Graduates will also be able to communicate their solution to others, including why and how a solution solves the problem and what assumptions were made.

6. **Ability to work effectively in a team.**
   Graduates were involved in at least one substantial team project. In most cases, this experience was a software development project, but other experiences would also appropriate in particular circumstances. Such projects challenged students by being integrative, requiring evaluation of potential solutions, and requiring work on a larger scale than typical course
projects. Students had opportunities to develop their interpersonal communication skills as part of their project experience.

7. **Commitment to life-long learning, and professional and ethical responsibility.**

Graduates realize that the computing field advances at a rapid pace, and graduates must possess a solid foundation that allows and encourages them to maintain relevant skills as the field evolves. Specific languages and technology platforms change over time. Therefore, graduates realize that they must continue to learn and adapt their skills throughout their careers. To develop this ability, students were exposed to multiple programming languages, tools, paradigms, and technologies as well as the fundamental underlying principles throughout their education. In addition, graduates were expected to manage their own career development and advancement. Graduates will seek career advancement by engaging in professional development activities, such as certifications, management training, or obtaining domain-specific knowledge.

Graduates recognize the social, legal, ethical, and cultural issues inherent in the discipline of computing. They will further recognize that social, legal, and ethical standards vary internationally. They are knowledgeable about the interplay of ethical issues, technical problems, and aesthetic values that play an important part in the development of computing systems. They will understand their individual and collective responsibility and the possible consequences of failure. They will understand their own limitations as well as the limitations of their tools.

8. **Effective communication and organizational skills.**

Graduates can make effective presentations to a range of audiences about technical problems and their solutions. This may involve face-to-face, written, or electronic communication. They are prepared to work effectively as members of teams. Graduates manage their own learning and development, including managing time, priorities, and progress.

9. **Awareness of the broad applicability of computing.**

Platforms range from embedded micro-sensors to high-performance clusters and distributed clouds. Computer applications impact nearly every aspect of modern life. Graduates understand the full range of opportunities available in computing.

10. **Appreciation of domain-specific knowledge.**

Graduates understand that computing interacts with many different domains. Solutions to many problems require both computing skills and domain knowledge. Therefore, graduates are able to communicate with, and learn from, experts from different domains throughout their careers.

References