|  |  |
| --- | --- |
| For Academic Affairs and Research Use Only | |
| CIP Code: |  |
| Degree Code: |  |

**New Course Proposal Form**

**[X] Undergraduate Curriculum Council**

**[ ] Graduate Council**

|  |
| --- |
| **[X] New Course or [ ]Experimental Course (1-time offering) (Check one box)** |

Signed paper copies of proposals submitted for consideration are no longer required. Please type approver name and enter date of approval.

Email completed proposals to [curriculum@astate.edu](mailto:curriculum@astate.edu) for inclusion in curriculum committee agenda.

|  |  |
| --- | --- |
| Edward Hammerand 9/24/2017 **Department Curriculum Committee Chair** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enter date…  **COPE Chair (if applicable)** |
| Hung-Chi Su 9/24/2017 **Department Chair:** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enter date…  **Head of Unit (If applicable)** |
| David F. Gilmore 9/29/2017 **College Curriculum Committee Chair** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enter date…  **Undergraduate Curriculum Council Chair** |
| Anne A. Grippo 9/29/2017 **College Dean** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enter date…  **Graduate Curriculum Committee Chair** |
| |  |  | | --- | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Enter date |   **General Education Committee Chair (If applicable)** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enter date…  **Vice Chancellor for Academic Affairs** |

1. Contact Person (Name, Email Address, Phone Number)

Hung-Chi Su, suh@astate.edu, (870)680-8119

2. Proposed Starting Term and Bulletin Year

Spring, 2018

3. Proposed Course Prefix and Number (Confirm that number chosen has not been used before. For variable credit courses, indicate variable range. *Proposed number for experimental course is 9*. )

CS 3613

4. Course Title – if title is more than 30 characters (including spaces), provide short title to be used on transcripts. Title cannot have any symbols (e.g. slash, colon, semi-colon, apostrophe, dash, and parenthesis). Please indicate if this course will have variable titles (e.g. independent study, thesis, special topics).

Web Application Development

5. Brief course description (40 words or fewer) as it should appear in the bulletin.

Introduction to the fundamentals, design patterns, interfaces, and technologies underlying web application development in a multi-tiered enterprise environment.

6. Prerequisites and major restrictions. (Indicate all prerequisites. If this course is restricted to a specific major, which major. If a student does not have the prerequisites or does not have the appropriate major, the student will not be allowed to register).

1. **Yes** Are there any prerequisites?
   1. If yes, which ones?

C or better in CS2114

* 1. Why or why not?

The material covered by the course requires some experience in computer programming, and an understanding of structured programming concepts, with some exposure to abstract data types.

1. **No** Is this course restricted to a specific major?
   1. If yes, which major? Enter text...

7. Course frequency(e.g. Fall, Spring, Summer). *Not applicable to Graduate courses.*

spring

8. Will this course be lecture only, lab only, lecture and lab, activity, dissertation, experiential learning, independent study, internship, performance, practicum, recitation, seminar, special problems, special topics, studio, student exchange, occupational learning credit, or course for fee purpose only (e.g. an exam)? Please choose one.

Lecture only

9. What is the grade type (i.e. standard letter, credit/no credit, pass/fail, no grade, developmental, or other [please elaborate])

Standard letter

10. **No** Is this course dual listed (undergraduate/graduate)?

11. **No** Is this course cross listed?

*(If it is, all course entries must be identical including course descriptions. Submit appropriate documentation for requested changes. It is important to check the course description of an existing course when adding a new cross listed course.)*

**11.1** – If yes, please list the prefix and course number of cross listed course.

Enter text...

**11.2** – **Yes / No** Are these courses offered for equivalent credit?

Please explain. Enter text...

12. **No** Is this course in support of a new program?

a. If yes, what program?

Enter text...

13. **No** Does this course replace a course being deleted?

a. If yes, what course?

Enter text...

14. **No** Will this course be equivalent to a deleted course?

a. If yes, which course?

Enter text...

15. **Yes** Has it been confirmed that this course number is available for use?

*If no: Contact Registrar’s Office for assistance.*

16. **No** Does this course affect another program?

If yes, provide confirmation of acceptance/approval of changes from the Dean, Department Head, and/or Program Director whose area this affects.

Enter text...

**Course Details**

17. Outline (The course outline should be topical by weeks and should be sufficient in detail to allow for judgment of the content of the course.)

Week 1 HTML5, CSS, the Document Object Model

Week 2 a static web server, and client/server communications

Week 3 The PHP language - syntax, semantics

Week 4 introduction to procedural and object-oriented features, and libraries

Week 5 Data persistence (flat-file storage) in PHP

Week 6 Client-side Javascript - syntax, semantics, object model, and interaction with the DOM

Week 7 Introduction to the MVC design pattern

Week 8 Database-based persistence, DBMS's, the Relational database model

Week 9 basic SQL queries, PHP's PDO, SQLite3

Week 10 Client-side Javascript - Investigation of using Javascript in data validation and secure AJAX transactions

with the server.

Week 11 Asynchronous client/server communication

Week 12 interactive and responsive interfaces

Week 13 Securing access using TLS; overview of public-key cryptography; secure password management techniques

Week 14 Survey of cloud deployment options; discussion of product scalability issues

18. Special features (e.g. labs, exhibits, site visitations, etc.)

None

19. Department staffing and classroom/lab resources

Enter text...

1. Will this require additional faculty, supplies, etc.?

No. The course has been in the rotation for several semesters as a temporary course. Its addition to the bulletin as a regular course will have no impact on department staffing or resources.

20. **No** Does this course require course fees?

*If yes: please attach the New Program Tuition and Fees form, which is available from the UCC website.*

**Course Justification**

21. Justification for course being included in program. Must include:

a. Academic rationale and goals for the course (skills or level of knowledge students can be expected to attain)

Applications based “in the Cloud” (also referred to as “Software as a Service” (SaaS)), as a new paradigm of information technology, has been widely embraced by industry. It represents a fundamental shift away from the previous “Software as a Platform” (SaaP) model. It is necessary to prepare students majoring in Computer Science (especially in the BA-track) to be professional application developers on cloud computing platforms. Goals Include: Experience with the multi-tiered development model inherent to web-based software. Understanding of client/server architectures. Experience with industry best-practices, design patterns, and modern language idioms. In-depth exploration of the “Model-View-Controller” design pattern. Awareness of security issues, and exploration of modern security best-practices including hashing and encryption schemes. Development of software applications that are “cloud”-based.

b. How does the course fit with the mission established by the department for the curriculum? If course is mandated by an accrediting or certifying agency, include the directive.

One part of the mission is to maintain the curriculum with updated technologies. The state of the art in the software industry has been shifting toward Software as a Service (SaaS) and mobile cloud applications in recent years; our students need to be prepared for this new development paradigm. This course provides students with experience in developing applications in a modern heterogeneous, multi-tiered software stack. This will give students better insights into the current state of industry than native-platform focused programming courses alone.

c. Student population served.

Undergraduate.

d. Rationale for the level of the course (lower, upper, or graduate).

The upper level of the course corresponds to the requirement that students already be experienced in the fundamentals of programming and ready to learn about applications of that knowledge in a modern enterprise environment.

**Assessment**

**University Outcomes**

22. Please indicate the university-level student learning outcomes for which this new course will contribute. Check all that apply.

|  |  |  |
| --- | --- | --- |
| * 1. **[ ]** Global Awareness | * 1. **[X ]** Thinking Critically | * 1. **[X ]** Information Literacy |

**Relationship with Current Program-Level Assessment Process**

23. What is/are the intended program-level learning outcome/s for students enrolled in this course? Where will this course fit into an already existing program assessment process?

1. ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
2. ability to function effectively on teams to accomplish a common goal.
3. use current techniques, skills, and tools necessary for computing practice.

24. Considering the indicated program-level learning outcome/s (from question #23), please fill out the following table to show how and where this course fits into the program’s continuous improvement assessment process.

*For further assistance, please see the ‘Expanded Instructions’ document available on the UCC - Forms website for guidance, or contact the Office of Assessment at 870-972-2989.*

|  |  |
| --- | --- |
| **Program-Level Outcome 1 (from question #23)** | ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs |
| Assessment Measure | The complete project will be used as a direct assessment of the outcome; employer surveys will be conducted as well. |
| Assessment  Timetable | The complete project will be evaluated in each offering of the course, reviewed annually, and reported on every three years; employer surveys will be conducted each fall and reported on every four years. |
| Who is responsible for assessing and reporting on the results? | Course instructor in coordination with the department assessment committee. |

|  |  |
| --- | --- |
| **Program-Level Outcome 2 (from question #23)** | ability to function effectively on teams to accomplish a common goal |
| Assessment Measure | Team members will be assessed during group meetings and project presentations; employer surveys will be conducted as well. |
| Assessment  Timetable | Team member performance will be evaluated in each offering of the course, reviewed annually, and reported on every three years; employer surveys will be conducted each fall and reported on every four years. |
| Who is responsible for assessing and reporting on the results? | Course instructor in coordination with the department assessment committee. |

|  |  |
| --- | --- |
| **Program-Level Outcome 3 (from question #23)** | use current techniques, skills, and tools necessary for computing practice |
| Assessment Measure | The complete project will be used as a direct assessment of the outcome; employer surveys will be conducted as well. |
| Assessment  Timetable | The complete project will be evaluated in each offering of the course, reviewed annually, and reported on every three years; employer surveys will be conducted each fall and reported on every four years. |
| Who is responsible for assessing and reporting on the results? | Course instructor in coordination with the department assessment committee. |

*(Repeat if this new course will support additional program-level outcomes)*

**Course-Level Outcomes**

25. What are the course-level outcomes for students enrolled in this course and the associated assessment measures?

|  |  |
| --- | --- |
| **Outcome 1** | Students will be able to create appropriate data models and application logic to implement a Software-as-a-Service (SaaS) application based on the Model-View-Controller paradigm within a client-server model to create scalable, maintainable applications. |
| Which learning activities are responsible for this outcome? | Lectures, reading assignments, hands-on programming assignments. |
| Assessment Measure | Software developed by students will be assessed at three levels, each scored according to a rubric: • Peer review: students will review selected applications from other students in the class. • Self-Assessment: students will assess the success of their software project on their own, based on a rubric. • Instructor Assessment: the instructor will assess the overall quality of the student’s project based on a rubric. |

|  |  |
| --- | --- |
| **Outcome 2** | Students will be able to identify and discuss ethical and security implications of cloud-based applications, and the responsibilities these place on the software developer. |
| Which learning activities are responsible for this outcome? | Lectures, in-class guided discussions about relevant industry news, reading assignments. The hands-on programming project will require attention to security details. |
| Assessment Measure | Student projects will be assessed for security risk as part of the Instructor’s Rubric (see Outcome (1) above). Discussion questions will be placed on the final exam to assess awareness of ethical and security concerns. |

*(Repeat if needed for additional outcomes)*

**Bulletin Changes**

|  |
| --- |
| **Instructions** |
| **Please visit** [**http://www.astate.edu/a/registrar/students/bulletins/index.dot**](http://www.astate.edu/a/registrar/students/bulletins/index.dot) **and select the most recent version of the bulletin. Copy and paste all bulletin pages this proposal affects below. Follow the following guidelines for indicating necessary changes.**  **\*Please note: Courses are often listed in multiple sections of the bulletin. To ensure that all affected sections have been located, please search the bulletin (ctrl+F) for the appropriate courses before submission of this form.**  - Deleted courses/credit hours should be marked with a red strike-through (~~red strikethrough~~)  - New credit hours and text changes should be listed in blue using enlarged font (blue using enlarged font).  - Any new courses should be listed in blue bold italics using enlarged font (***blue bold italics using enlarged font***)  *You can easily apply any of these changes by selecting the example text in the instructions above, double-clicking the ‘format painter’ icon 🡪 , and selecting the text you would like to apply the change to.*  *Please visit* [*https://youtu.be/yjdL2n4lZm4*](https://youtu.be/yjdL2n4lZm4) *for more detailed instructions.* |

**2017-2018 Undergraduate Bulletin**

**Page 565**

CS 3113. Algorithms and Advanced Data Structures Analysis of data structures and associated algorithms. Examination of advanced tree structures, heaps, hashing techniques, and graph algorithms. Prerequisites, C or better in CS 2124 and MATH 2183, and MATH 2204 or MATH 2143 or MATH 2194. Fall, Spring.

CS 3123. Programming Languages Survey of organization and behavior of programming languages. Examination of data typing, control structures, syntactic representation and specification. Prerequisites, CS 2124. Spring.

CS 3223. Computer Organization Basic principles of computer architectural design including instruction set principles, pipelining, instruction level parallelism, memory hierarchy, storage systems, and multiprocessing. Prerequisites, MATH 2204 or MATH 2143 or MATH 2194 and CS 2114. Fall.

CS 3233. Operating Systems Policies, design issues, and implementation techniques for operating system software. Synchronization, process scheduling, memory and storage manage-ment, and system protection. Prerequisite, CS 3113. Spring.

***CS 3613. Web Application Development Introduction to the fundamentals, design patterns, interfaces, and technologies underlying web application development in a multi-tiered enterprise environment. Prerequisite, C or better in CS 2114. Spring.***

CS 4113. Software Engineering Techniques of design, implementation, automated tools, quality assurance, metrics, and maintenance for large scale software systems. Projects include team programming experience. Prerequisite, CS 3113. Fall.