PURDUE UNIVERSITY
REQUEST FOR ADDITION, EXPIRATION,
OR REVISION OF AN UNDERGRADUATE COURSE
(10000-40000 LEVEL)

DEPARTMENT: Computer Science
EFFECTIVE SESSION: Fall 2014 (2015-16)

INSTRUCTIONS: Please check the items below which describe the purpose of this request.

☐ 1. New course with supporting documents
☐ 2. Add existing course offered at another campus
☐ 3. Expiration of a course
☐ 4. Change in course number
☐ 5. Change in course title
☐ 6. Change in course credit type
☐ 7. Change in course attributes (department head signature only)
☐ 8. Change in instructional hours
☐ 9. Change in course description
☐ 10. Change in course requisites
☐ 11. Change in semesters offered (department head signature only)
☐ 12. Transfer from one department to another

PROPOSED:

Subject Abbreviation: CS
Course Number: 22800
Credit Title: C/C++ Programming for Electrical and Computer Engineering
Offered: Summer

EXISTING:

Subject Abbreviation: CS
Course Number: 22800
Credit Title: C/C++ Programming
Offered: Fall

TERMS OFFERED:

☐ Summer ☐ Fall ☐ Spring

CAMPUS (ES) INVOLVED:

☐ Calumet ☐ East Chicago
☐ Cont Ed ☐ Ft. Wayne
☐ Indiana Dewey ☐ West Lafayette

CREDIT TYPE:

1. Fixed Credit: Cr. Hrs: 4.0
2. Variable Credit Range: 3. Minimum Cr. Hrs: 3.0
3. Maximum Cr. Hrs: 3.0

COURSE ATTRIBUTES:

☐ Pass/No Pass Only ☐ 5. Registration Approval Type:
☐ Satisfactory/Unsatisfactory Only ☐ Department
☐ Repeatable ☐ Instructor
☐ Maximum Repeatable Credit: 3.0
☐ Credit by Examination: 8. Honors
☐ Fee: ☐ Coop ☐ Lab ☐ Rate Request

Schedule Type:

Lecture: 75/2/16 0.75
Lab/Prac: 75/1/16 0.25

COURSE DESCRIPTION (INCLUDE REQUISITES/RESTRICTIONS):

Pre-req: ENGR 12800 or consent of instructor.

This course will introduce the programming in C and fundamentals of object-oriented programming in C++ to solve engineering problems. This course introduces the C programming language features in the beginning of the semester. Topics covered include data types, control structures, standard input/output, file input/output, mathematical library, problem-solving, functions, arrays, dynamic memory, and pointers. Then the course provides an introduction to object-oriented programming using C++ language. Students will gain understanding of data abstractions, classes, inheritance, composition, method overloading and overriding, generic programming, and standard template library. Students are expected to complete lab assignments and several programming assignments.

COURSE LEARNING OUTCOMES:

The goal of this course is the introduction of C++ programming languages to solve engineering problems. Specific learning outcomes are listed below. The letters in parentheses refer to ABET Program Learning Outcomes. A student who successfully fulfills the course requirements will have demonstrated:

1. An ability to use C programming language to solve elementary engineering problems. (a, g, h, k, i) 2. An ability to use data types, variables, and arithmetic operators. (a, e, h) 3. An ability to use conditional statements and loops structures. (a, e, h, k) 4. An understanding of the use of arrays and pointers. (a, h, k) 5. An ability to use the dynamic memory. (c, h, k) 6. An ability to use mathematical library. (c, h, k) 7. An ability to develop function-oriented programs. (a, e, h, k) 8. An understanding of the passing arguments to/from functions. (a, e, h, k) 9. An ability to use standard input/output and file input/output operations. (a, e, h, k) 10. An ability to use object-oriented programming in C++ to solve basic engineering problems. (a, b, e, h, k) 11. An understanding of the use of classes and access control to class members. (c, k) 12. An ability to use class inheritance and composition. (c, h, k) 13. An understanding of the method overloading and overriding. (g, h, k) 14. An ability to use generic programming and standard template library. (c, h, k)

[Signatures and dates of approval]
1. Course Description

Pre-req: ENGR 128 or consent of instructor.

This course will introduce the programming in C and fundamentals of object-oriented programming in C++ to solve engineering problems. This course introduces the C programming language features in the beginning of the semester. Topics covered include data types, control structures, standard input/output, file input/output, mathematic library, problem-solving, functions, arrays, dynamic memory, and pointers. Then the course provides an introduction to object-oriented programming using C++ language. Students will gain understanding of data abstractions, classes, inheritance, composition, method overloading and overriding, generic programming, and standard template library. Students are expected to complete lab assignments and several programming assignments.

2. Course Goals & Course Learning Outcomes

The goal of this course is the introduction of C/C++ programming languages to solve engineering problems. Specific learning outcomes are listed below. The letters in parentheses refer to ABET Program Learning Outcomes. A student who successfully fulfills the course requirements will have demonstrated:

1. An ability to use C programming language to solve elementary engineering problems. (a, c, e, k)
2. An ability to use data types, variables, and arithmetic operators. (a, e)
3. An ability to use conditional statements and loops structures. (c, e, k)
4. An understanding of the use of arrays and pointers. (e, k)
5. An ability to use the dynamic memory. (c, k)
6. An ability to use mathematic library. (e, k)
7. An ability to develop function-oriented programs. (c, e, k)
8. An understanding of the passing arguments to/from functions. (c, k)
9. An ability to use standard input/output and file input/output operations. (c, k)
10. An ability to use object-oriented programming in C++ to solve basic engineering problems. (a, c, e, k)
11. An understanding of the use of classes and access control to class members. (c, k)
12. An ability to use class inheritance and composition. (c, k)
13. An understanding of the method overloading and overriding. (j, k)
14. An ability to use generic programming and standard template library. (c, k)

3. Book

3.1 Textbook

3.2 Reference books


4. Tentative Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
<th>Lab Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to C program, Visual Studio.NET environment, C program structure, Preprocessor directives, Data types, Constants, Variables</td>
<td>“Hello world” on Visual Studio.NET</td>
</tr>
<tr>
<td>2</td>
<td>Arithmetic operators, Standard Input/Output, I/O Formatting, Structured programming</td>
<td>C program structure, Variables, Arithmetic operators</td>
</tr>
<tr>
<td>3</td>
<td>Conditional statements, Loops structures</td>
<td>Standard Input/Output, I/O Formatting</td>
</tr>
<tr>
<td>4</td>
<td>File Input/Output, Mathematical functions, Modular programming</td>
<td>Structural programming, Control structures</td>
</tr>
<tr>
<td>5</td>
<td>User-defined functions</td>
<td><strong>Midterm Exam I</strong></td>
</tr>
<tr>
<td>6</td>
<td>Arguments passing among functions, Scope, Recursion</td>
<td>File Input/Output</td>
</tr>
<tr>
<td>7</td>
<td>Arrays and matrices</td>
<td>Function-oriented programming</td>
</tr>
<tr>
<td>8</td>
<td>Sorting, Searching, Pointers</td>
<td>Array</td>
</tr>
<tr>
<td>9</td>
<td>Pointer Arguments</td>
<td>Pointers</td>
</tr>
<tr>
<td>10</td>
<td>Dynamic memory</td>
<td><strong>Midterm Exam II</strong></td>
</tr>
<tr>
<td>11</td>
<td>Introduction to object-oriented programming, C++ program structure, Classes, Constructor</td>
<td>Dynamic memory</td>
</tr>
<tr>
<td>12</td>
<td>Access control to class members, Developing OO application</td>
<td>Intro to C++ programming</td>
</tr>
<tr>
<td>13</td>
<td>Inheritance &amp; composition</td>
<td>OO programming</td>
</tr>
<tr>
<td>14</td>
<td>Generic programming, Standard template library</td>
<td>Inheritance &amp; composition</td>
</tr>
<tr>
<td>15</td>
<td>Method overloading &amp; overriding</td>
<td>Standard template library</td>
</tr>
</tbody>
</table>

**Final Exam**