# Purdue University
REQUEST FOR ADDITION, EXPIRATION, OR REVISION OF AN UNDERGRADUATE COURSE (100-400 LEVEL)

**RTMNT:** ECE  
**EFFECTIVE SESSION:** Summer 2008

<table>
<thead>
<tr>
<th>R e a s o n s</th>
<th>P r o p o s e d:</th>
<th>E x i s t i n g:</th>
<th>T e r m s O f f e r e d:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. New course with supporting documents</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Add existing course offered at another campus</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Expiration of a course</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Change in course number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Change in course title</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Change in course credit/type</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Subject Abbreviation:** ECE  
**Course Number:** 435

**C O U R S E  T A B L E:**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Satisfactory/Unsatisfactory Only</td>
<td>8. Variable Title</td>
<td></td>
</tr>
<tr>
<td>3. Repeatable</td>
<td>9. Remedial</td>
<td></td>
</tr>
<tr>
<td>4. Credit by Examination</td>
<td>10. Honors</td>
<td></td>
</tr>
<tr>
<td>5. Designator/Required</td>
<td>11. Full Time Privilege</td>
<td></td>
</tr>
<tr>
<td>6. Special Fees</td>
<td>12. Off Campus Experience</td>
<td></td>
</tr>
</tbody>
</table>

**C O U R S E  D E S C R I P T I O N  (I N C L U D E  R E Q U I S I T E S):**


**C R O S S - L I S T E D  C O U R S E S:**

**C A L U M E T  D E P A R T M E N T  H E A D:** Date  
**C A L U M E T  S C H O O L  D E A N:** Date  
**F O R T  W A Y N E  D E P A R T M E N T  H E A D:** Date  
**F O R T  W A Y N E  S C H O O L  D E A N:** Date  
**I N D I A N A P O L I S  D E P A R T M E N T  H E A D:** Date  
**I N D I A N A P O L I S  S C H O O L  D E A N:** Date  
**N O R T H  C E N T R A L  D E P A R T M E N T  H E A D:** Date  
**N O R T H  C E N T R A L  C H A N C E L L O R:** Date  
**W E S T  L A F A Y E T T E  D E P A R T M E N T  H E A D:** Date  
**W E S T  L A F A Y E T T E  C O L L E G E / S C H O O L  D E A N:** Date  
**W E S T  L A F A Y E T T E  R E G I S T R A R:** Date
### Request for Addition, Expiration, or Revision of an Undergraduate Course

**Department:** ECE  
**Effective Session:** Summer 2008

**PURDUE UNIVERSITY**

**Request for Addition, Expiration, or Revision of an Undergraduate Course**

1. **Proposed:**
   - Subject Abbreviation: ECE
   - Course Number: 435
   - Long Title: Object-Oriented Design Using C++ and Java
   - Short Title: (Abbreviated title will be entered by the Office of the Registrar if limited to 22 characters only)

2. **CREDIT TYPE:**
   - Fixed Credit: 3.0

3. **Instructional Type:**
   - Lecture: 50 minutes per mg
   - Recitation: 3 meetings per week
   - Laboratory: 16 weeks

4. **COURSE ATTRIBUTES:**
   - Pass/No Pass Only
   - Satisfactory/Unsatisfactory Only
   - Repeatable
   - Maximum Repeatable Credit: 3.0
   - Designator Required
   - Cross-Listed Courses

5. **Course Description (Include Requisites):**

---

**Department Head Signatures:**

- Calumet Department Head
- Fort Wayne Department Head
- Indianapolis Department Head
- North Central Department Head
- West Lafayette Department Head

- Calumet School Dean
- Fort Wayne School Dean
- Indianapolis School Dean
- North Central Chancellor
- West Lafayette College/School Dean

**Office of the Registrar**
TO: The Faculty of the College of Engineering  
FROM: The Faculty of the School of Electrical and Computer Engineering  
RE: New Undergraduate Level Course ECE 435  

The Faculty of the School of Electrical and Computer Engineering has approved the following new course. This action is now submitted to the Engineering Faculty with a recommendation for approval.

ECE 435 **Object-Oriented Design Using C++ and Java**

Sem. 2, Class 3, cr. 3.  
Prerequisite: ECE 462 and Consent of Instructor  


Reason: It is now widely recognized that just knowing Object Oriented (OO) languages and having access to a library of classes is not sufficient for creating OO designs. This realization has led to the emergence of a “patterns movement” in the OO community. Patterns are the “best practice” designs that have evolved over the years for tackling issues such as how to make objects sharable; how and when to assign responsibilities to objects; how to make the OO design reusable in other similar contexts, etc.

Mark J. T. Smith, Head  
School of Electrical and Computer Engineering

**APPROVED FOR THE FACULTY OF THE SCHOOLS OF ENGINEERING BY THE ENGINEERING CURRICULUM COMMITTEE**

ECC Minutes #9  
Date 11/26/07  
Chairman ECC
Supporting Documentation

Level: Undergraduate Level
Course Instructor: Avinash C. Kak

Course Outline:

Topics                                      Lectures
1. Course Introduction                      1
2. Software Development Process for Large OO Programs  1
3. Use Cases, Class Diagrams                1
4. Class Diagrams (Advanced Concepts)       1
5. Interaction, Package, State, and Activity Diagrams  3
6. Extending Classes in C++ and Java         5
7. OO Design using Multiple Inheritance in C++  4
8. Design Patterns                          6
9. OO for GUI Design with Java, C++, and C  
   (AWT/Swing in Java, Qt in C++, and GNOME/GTK+ in C)  7
10. OO Design using Multithreading          4
11. OO Design in Network Programming        4
12. Design for Database Programming         3
13. OO Design for Web Services Programming  2
14. Exams 2                                  2
Total                                       44

Text(s):

Recommended Reference(s):

C++ Programming Language, 3rd edition, B. Stroustrup, Addison-Wesley, 1997,


Outcomes:
A student who successfully fulfills the course requirements will have demonstrated:

i. a knowledge of the Unified Modeling Language for the conceptual design of object-oriented programs. [3,4;e,k]

ii. an ability to design object-oriented solutions to programming problems using previously developed “best practice” design components. [3,4;e,k]

iii. an understanding of the pros and cons associated with multiple inheritance in C++. [3,4;e,k]

iv. a knowledge of graphics and user interface programming with Java. [3,4;e,e,k]

v. an ability to carry out databases programming in Java. [3,4;e,k]

vi. an understanding of multithreading issues in Java. [3,4;e,k]

Assessment Methods for Course Outcomes: Each of the outcomes will be assessed by giving the students appropriate C++ and Java programming assignments.