JOINT INSTITUTE
交大密西根学院

Standard Undergraduate Course Profile

<table>
<tr><td>COURSE NUMBER: Ve280</td><td>COURSE TITLE: Programming and Introductory Data Structures</td></tr>
<tr><td>CREDIT: 4</td><td>PREREQUISITES: Vg101 or equivalent</td></tr>
<tr><td>TEXTBOOKS/REQUIRED MATERIAL:
“Problem Solving with C++, 8th Edition,” W. Savitch</td><td>PREPARED BY: Weikang Qian
LAST UPDATED: May 25, 2012
DATE OF DISCIPLINE GROUP APPROVAL:
DATE OF UC APPROVAL:</td></tr>
<tr><td>CATALOG DESCRIPTION (No more than 100 words):
Techniques for algorithm development and effective programming; Testing and program correctness; Program language syntax and static and runtime semantics; Procedure abstraction, recursion, and parameter passing methods; Abstract data type, inheritance, template, and polymorphism; Structured data types, pointers, arrays, linked data structures, stacks, and queues.</td><td>COURSE TOPICS:
1. Linux basics and compiling program on Linux
2. Review of C++ Basics, such as array, pointer, etc.
3. Procedural abstraction, function call mechanism, and recursion
4. Function pointer
5. Enum
6. Program arguments
7. Testing
8. Debugging
9. IO
10. Exception
11. Abstract data type and class
12. Inheritance and virtual function
13. Interface (i.e., abstract base class)
14. Representation invariant
15. Dynamic memory allocation and dynamic arrays
16. Overloaded constructor, destructor, copy constructor, and overloaded assignment operator
17. Operator overloading and friend mechanism
18. Linked list (including linked list traversal)
19. Stack and queue
20. Polymorphism, template, and STL</td></tr>
<tr><td colspan="2">COURSE STRUCTURE and CONTACT HOUR: 48 hours of lecture and 12 hours of demo/discussion</td></tr>
<tr><td>COURSE OUTCOMES [Student Outcomes* in brackets]

for each course outcome, links to the Student Outcomes are identified in brackets.</td><td>After completing Ve280, students should be able to:
1. Take a problem and consider various possible approaches for solving it. [2,6]
2. Select an approach—or algorithm—that provides for a simple, clean, well-structured solution. [2]
3. Convert the algorithm into C++ code, using good design and style. [1,2]
4. Develop and debug a program on Linux operating systems. [1,2]
5. Test and debug the program using rigorous techniques. [1,2]
6. Understand the concepts of top-down design, data encapsulation, information hiding, and procedural and data abstraction. [1]
7. Design, implement, and use classes, including constructors, destructors, and operator overloading. [1,2]
8. Implement dynamic data structures for stacks, queues, and lists. [1]
9. Be able to quickly design, implement, test, and debug a large scale project independently (1000+ lines of code). [1,2]</td></tr>
<tr><td>COURSE OBJECTIVES [Course Outcomes in brackets]

for each course objective, links to the course outcomes are identified in brackets.</td><td>1. To give an introduction to programming and to provide a foundation on data structures. [1, 6, 7, 8]
2. To provide students with experience on how to design and implement an algorithm to solve a practical problem. [1, 2, 3, 8, 9]
3. To teach students some useful techniques for developing, debugging, and testing programs. [4, 5, 9]</td></tr>
</table>

ASSESSMENT TOOLS [Course Outcomes in brackets] *for each assessment tool, links to the course outcomes are identified*	Programming Projects [1, 2, 3, 4, 5, 6, 7, 8, 9] Midterm and Final Exam [1, 2, 3, 5, 6, 7, 8]

Rev. 1: July 2020

ABET Student Outcomes* —— Apply to Engineering, Math, and Science Courses Only

1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors

3) an ability to communicate effectively with a range of audiences

4) an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts

5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies