Faculty members are required to have the outline submitted to the Academic Affairs Office. The course outline is the form used for approval of new courses by the Academic Affairs and Standards Council.

**DEPT.**  CSCI  **COURSE NUMBER:**  2245

**NUMBER OF CREDITS:**  4  **Lecture:**  4  **Lab:**  0

**Course Title:**  
Fundamentals of Programming II

**Catalog Description:**  
Discusses topics including object-oriented programming techniques, essential data structures such as stacks, queues, trees, sorting and searching algorithms using a high-level programming language.

**FULFILLS MN TRANSFER CURRICULUM AREA(S)**

Goal 1: Communication: ____ by meeting the following competencies:

Goal 2: Critical Thinking: ____ by meeting the following competencies:

Goal 3: Natural Sciences: ____ by meeting the following competencies:

Goal 4: Mathematics/Logical Reasoning: ____ by meeting the following competencies:

Goal 5: History and the Social and Behavioral Sciences: ____ by meeting the following competencies:

Goal 6: The Humanities and Fine Arts: ____ by meeting the following competencies:

Goal 7: Human Diversity: ____ by meeting the following competencies:

Goal 8: Global Perspective: ____ by meeting the following competencies:

Goal 9: Ethical and Civic Responsibility: ____ by meeting the following competencies:

Goal 10: People and the Environment: ____ by meeting the following competencies:

**Prerequisites or Necessary Entry Skills/Knowledge:**

CSCI 2240 – Fundamentals of Programming I
Topics to be Covered:

- Pointers
- Operator overloading
- Inheritance
- Polymorphism
- Stream input/output
- Exception handling
- File processing
- Searching and sorting
- Linked lists
- Stacks, queues, trees

Student Learning Outcomes

1) Develop and implement correct and efficient programs using the C++ language.
2) Define, compare and contrast the fundamental concepts of object-oriented programming: data abstraction, encapsulation, inheritance and polymorphism.
3) Design algorithms according to object-oriented concepts.
4) Design and develop classes which implement the concepts of data abstraction, encapsulation, inheritance and polymorphism.
5) Design and develop programs implementing data structures utilizing the Standard Template Library.
6) Implement exception handling.
7) Examine searching and sorting algorithms.
8) Define the finer points of pointers, dynamic allocation, linked list, stacks, queues, trees.

Is this course part of a transfer pathway: Yes ☐ No ☒

Revised Date: May 2020