RARITAN VALLEY COMMUNITY COLLEGE
ACADEMIC COURSE OUTLINE

GDEV 242 Object Oriented Programming

I. Basic Course Information

    A. Course Number and Title: **GDEV 242 Object Oriented Programming**

    B. New or Modified Course: **Modified**

    C. Date of Proposal: Semester: **Fall** Year:**2018**

    D. Effective Term: **Fall 2019**

    E. Sponsoring Department: **Visual and Performing Arts**

    F. Semester Credit Hours: 3

    G. Weekly Contact Hours   4   Lecture: 2
    Laboratory: 2
    Out of class student work per week: 5

    H. Prerequisites/Corequisites:
        CSIT 105 Foundations of Computer Science or
        CSIT 103 Computer Concepts & Programming

    I. Laboratory Fees: Yes

    J. Name and E-Mail Address of Department Chairs and Divisional Dean at time of
      Approval:
        John Sichel, john.sichel@raritanval.edu and Vandana Nadkarni,
        vandana.nadkarni@raritanval.edu, Department Chairs;
        Patrice Marks, Patrice.marks@raritanval.edu, Divisional Dean

II. Catalog Description

Prerequisite: CSIT-105 Foundations of Computer Science or CSIT-103 Computer Concepts & Programming. Building on the ideas covered in both CSIT-103 Computer Concepts and Programming and CSIT-105 Foundations of Computer Science, this course introduces the student to the concepts and features of Object-Oriented Programming (OOP). Course topics include encapsulation, inheritance, polymorphism, classes and objects, arrays, abstract classes and methods, message passing and unit testing.. Students will write programs

using one of the dominant object oriented languages in use such as, but not limited to Java, C#, C++ or Python.

III. Statement of Course Need

    A. The course is needed because object oriented design is the de facto design paradigm of professional software developers. It has proven to help manage software complexity, help extend the functionality of software systems, and help control the high cost of software maintenance. Employers and students have requested this course.

    B. This course has a lab component to allow students to develop programming skills using object oriented design principles.

    C. This course generally transfers as a replacement for the local course in object oriented programming, a computer science program elective or a free elective.

IV. Place of Course in College Curriculum

    A. Free Elective

    B. This course meets a program requirement for Game Development AAS

    C. This course is an option in:
1. Interactive Digital Media & Web Development AS
2. Interactive Digital Media & Web Development AAS
3. Interactive Digital Media & Web Development Certificate
4. Information Systems & Technology AS
5. Information Systems & Technology AAS

    D. Computer Elective on the Computer and Programming Electives List

    E. Programming Elective on the Computer and Programming Electives List

    F. To see course transferability: a) for New Jersey schools, go to the NJ Transfer website, www.njtransfer.org; b) for all other colleges and universities, go to the individual websites.

V. Outline of Course Content

    A. What is an Object
1. distinction between human cognition and specificity of detail required by computers
2. Place of object oriented languages in history of programming
3. review of standard control structures and data types

    B. What is the syntax used with objects
1. syntax for creating, instantiating and using objects
2. fundamentals of encapsulation, private, public and protected members
3. static member fields and functions
4. Object constructors, destructors

    C. How do objects interact with one another

1. Composition as a programming abstraction technique
2. Modularization of programming solutions
3. Message passing via method calls

D. Building collections of objects
   1. Using generic utility libraries for collections of objects like Vectors, Lists, HashMaps, Sets, etc,
   2. Iterating over collections
   3. Accessing a single object from a collection

E. Documenting and Testing software objects
   1. Standard documenting procedures for the language in use
   2. Unit testing and software development best practice

F. Partitioning objects for maximum reusability
   1. Addressing issues of coupling and cohesiveness in software code
   2. Coding to the Interface

G. Object inheritance trees
   1. Protected members
   2. Conversion and casting between types
   3. Polymorphism and virtual lookup tables (VLUTs)

H. Abstract and virtual objects
   1. Abstract classes
   2. Concrete classes
   3. Dynamic binding

I. Template based Classes
   1. Function templates
   2. Class templates
   3. Syntax used with template classes

J. Error handling for exceptional cases
   1. Try, throw, catch blocks

K. File Processing
   1. files and streams
      a) from the file system
      b) from the internet
   2. serialization of objects and persistence

VI. General Education and Course Learning Outcomes

A. General Education Learning Outcomes:

At the completion of the course, students will be able to:
1. The student will identify a problem and analyze it in terms of its significant parts, design an algorithm needed to solve the problem, and then construct code to implement and test the algorithm. (GE-NJ 2, 4 *)

* embedded critical thinking

B. **Course Learning Outcomes:**

At the conclusion of the course, students will be able to*:*
1. Define the key precepts of object oriented design
2. Apply control structures to write programs using an object oriented language according to object oriented design precepts.
3. Identify and use appropriate data structures in programs.
4. Analyze a problem, develop an object oriented system and justify the design choices made.

C. **Assessment Instruments**

1. laboratory products
2. art work
3. research papers
4. group projects
5. essays
6. journals – of code development via source code management system
7. portfolios
8. computer programs – in language of instruction

VII. Grade Determinants

A. Computer programs
B. Journals
C. Group projects
D. Exams

The primary formats, modes, and methods for teaching and learning that may be used in the course:
A. lecture/discussion
B. small-group work
C. computer-assisted instruction
D. guest speakers
E. laboratory
F. student oral presentations
G. simulation/role playing
H. student collaboration
I. independent study
J. Interpreting large code bases

VIII. Texts and Materials

.

A. suggested textbooks
Dependent upon language

Java: Objects First with Java | A Practical Introduction using BlueJ (6th ed.), David Barnes & Michael Kölling , Prentice Hall

C++: The C++ Programming Language, 4th Edition 4th Edition**,** Bjarne Stroustrop, Addison Wesley

C#: Beginning Object-Oriented Programming with C#, Jack Purdum, Wrox Inc.

B. interviews
C. journals
D. code reviews
E. code documentation
F. film and video
G. audio sources
H. web sources
I. other computer-based sources
(E-I) many computer scientists responsible for creating languages provide podcasts and video lectures about their creations. This can help students understand that the language they are using is a designed object, and a piece of software in its own right. These will be pulled from sites like Microsoft Site Developers Network, YouTube, Apple iTunes and other online sources as appropriate

(Please Note: The course outline is intended only as a guide to course content and resources. Do not purchase textbooks based on this outline. The RVCC Bookstore is the sole resource for the most up-to-date information about textbooks.)

IX. Resources

A. Computer lab with the appropriate Integrated Development Environment (IDE) installed for the language in use
B. Functioning audio and video drivers for individual playback of audio and video streams at student computer workstations
C. White board for written examples
D. Data Projector for displaying code, configured to display code directly on whiteboard

X. Honors Option
n/a