ANNEX 2 – COURSE DESCRIPTION

<table>
<tr><td>Course Title</td><td colspan="5">Numerical Methods</td></tr>
<tr><td>Course Code</td><td colspan="5">ACSC285</td></tr>
<tr><td>Course Type</td><td colspan="5">Elective</td></tr>
<tr><td>Level</td><td colspan="5">BSc (Level 1)</td></tr>
<tr><td>Year / Semester</td><td colspan="5">3rd (Spring)</td></tr>
<tr><td>Teacher’s Name</td><td colspan="5">Dr Savvas Pericleous</td></tr>
<tr><td>ECTS</td><td>6</td><td>Lectures / week</td><td>3</td><td>Laboratories/week</td><td>-</td></tr>
<tr><td>Course Purpose</td><td colspan="5">The primary purpose of the course is to revisit fundamental problems of calculus (such as, solving non-linear equations, integration and differentiation and solving first order ODEs) and examine numerical techniques for obtaining “as good as we want” approximate solutions, in cases when these problems cannot be solved analytically or solved exactly within a reasonable amount of time. Students are also expected to understand the limitations of floating point arithmetic, as well as, the effect of the nature of some problems and implement methods for minimizing the induced arithmetic error. Finally, students will compare three different techniques for polynomial interpolation.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="5">By the end of the course, students must be able to:
1. Identify the need for numerical methods and determine how they can be successfully applied to many important scientific problems that cannot be solved exactly within a reasonable amount of time.
2. Identify limitations and compromises inherent in numerical computation.
3. Examine the influence of the nature of the problem to be solved, understand the properties of floating-point arithmetic, the architecture of available computers and determine the effect of round off errors or loss of significance.
4. Analyse various methods for solving non-linear equations, including bracketing, bisection, Newton-Raphson, secant and iterative methods. Evaluate their appropriateness for different examples, and assess their robustness and accuracy, as well as their rate of convergence.
5. Acquire a basic knowledge of numerical approximation techniques (Taylor-series Method or Newton-Cotes Rules) for mathematical expressions, such as derivatives and definite integrals, and learn how, why, and when these techniques can be expected to work.
6. Evaluate various methods (Euler and Midpoint methods) for solving first-order Ordinary Differential Equation.
7. Compare a number of different polynomial interpolation techniques</td></tr>
</table>

<table>
<tr><td></td><td colspan="3">for Curve Fitting (Monomial Basis, Newton’s Divided-Difference and Lagrange Interpolating Polynomials) and illustrate their applicability.
8. Write simple programs for the proposed numerical algorithms in Matlab or other programming environments.</td></tr>
<tr><td>Prerequisites</td><td>AMAT122</td><td>Corequisites</td><td>None</td></tr>
<tr><td>Course Content</td><td colspan="3">• Tools for Scientific Computation: Mathematical background from Calculus, Taylor’s theorem and the Lagrange form of the remainder term, approximation of functions and derivatives; measuring and controlling errors.
• Errors in Computer Arithmetic: Floating point representation and arithmetic; rounding errors and its consequences.
• Solving non-linear equations: Iterative methods; bracketing and bisection method; Newton-Raphson & secant method; convergence rates and criteria.
• Curve Fitting: Polynomial Interpolation with Monomial Basis; Newton’s Divided-Difference Interpolating Polynomials; Lagrange Interpolating Polynomials; Spline Interpolation; Least Squares Regression Approximation
• Numerical Integration and Differentiation: Newton-Cotes Rules (Rectangular, Trapezoidal and Simpson’s). Forward, backward and central difference approximation methods for derivatives using Taylor’s theorem.
• Solving first-order Ordinary Differential Equations: Euler and Midpoint Methods for solving first order ODEs.</td></tr>
<tr><td>Teaching Methodology</td><td colspan="3">Students are taught the course through lectures (3 hours per week). For the delivery of the class material, power point presentations are primarily used, along with the whiteboard. The lecture notes, consisting of slides presented in class, the course outline and additional material, are made available to the students through the University’s e-learning platform. Students are also advised to use the subject’s textbook or reference books for further reading and practice in solving related exercises. The theoretical part of each lecture is accompanied with detailed solved examples on which emphasis is given in the class. The solutions to these exercises, as well as specimen solutions for all tests and assignments, are discussed with students. Students are encouraged to make full use of the instructor’s office hours (6 per week), where they can ask questions and further discuss lecture material on a one-to-one basis.</td></tr>
<tr><td>Bibliography</td><td colspan="3">(a) <u>Textbooks:</u>
• Steven Chapra and Raymond Canale, Numerical Methods for Engineers, 7th Edition, McGraw Hill, 2015

(b) <u>References</u>:
• J.H. Mathews, K.D. Fink. Numerical Methods using MATLAB, (Fourth Edition), Pearson Prentice Hall, 2004
• G. W. Recktenwald. Numerical Methods with MATLAB: Implementations and Applications, 2nd Edition, Pearson, 2001</td></tr>
</table>

Assessment	The Students are assessed via continuous assessment throughout the duration of the Semester, which forms the Coursework grade and the final written exam. The coursework and the final exam grades are weighted 40% and 60%, respectively, and compose the final grade of the course. The methods for the continuous assessment of the students, are primarily mid-term written tests, and assignments, The assessment weight, date and time of each type of continuous assessment is being set at the beginning of the semester via the course outline. An indicative weighted continuous assessment of the course coursework is shown below: • Assignments 25% • Mid-Term written exams 75% Students are prepared for the final exam, by revision on the taught material, problem solving and concept testing. The final assessment is designed to comply with the subject's expected learning outcomes.
Language	**English**