Course Syllabus

STI VISION: To be the leader in innovative and relevant education that nurtures individuals to become competent and responsible members of society. **STI MISSION:** We are an institution committed to provide knowledge through the development and delivery of superior learning systems. We strive to provide optimum value to all our stakeholders - our students, our faculty members, our employees, our partners, our shareholders, and our community. We will pursue this mission with utmost integrity, dedication, transparency, and creativity.	**STI INSTITUTIONAL OUTCOMES:** **Character (IO1):** An STIer is a person of character. An STIer takes responsibility for his/her actions, treats people with respect, and lives with integrity. **Critical Thinker (IO2):** An STIer is a critical thinker. An STIer challenges and analyses all information through sound questioning and is unafraid to push for creative ideas. **Communicator (IO3):** An STIer communicates to understand and be understood. An STIer discerns the value of information read or heard and effectively expresses his/her own emotions when sharing information, may it be spoken or written. **Change-adept (IO4):** An STIer is change-adept. An STIer can adjust, adapt, and reinvent continuously to changing circumstances. An STIer believes in letting go of the old and embracing the new to achieve his/her fullest potential.

SERIAL NUMBER: IT1808	**COURSE TITLE:** INTEGRATIVE PROGRAMMING 1	**CREDIT:** 2 units lec, 1 unit lab (2 hrs. lecture per week, 3 hrs. laboratory per week)
COURSE DESCRIPTION:	This course covers the introduction to C# language. The emphasis is to train students to design, implement, test, and debug programs intended to solve computing problems using fundamental programming constructs.	
PREREQUISITE:	COMPUTER PROGRAMMING 2	
COURSE OUTCOMES:	After successful completion of this course, the student should be able to: (ITPO01 – PI1, ITPO02 – PI1, ITPO05 – PI3, ITPO06 – PI3, ITPO07, - PI4 ITPO09 – PI6, ITPO10 – PI6) CO1. Apply basic language syntax and basic principles in programming; CO2. Design, implement, test, and debug a program based on a given specification that uses each of the following fundamental programming components: data types, basic computation, simple I/O, conditional and iterative structures, arrays, and strings. CO3. Implement object-oriented programming concepts in program development; and CO4. Determine and apply debugging techniques for solving errors and inhibiting program acceptance.	
MANDATED BOOK:	None	
REFERENCES:	1. Deitel, P. and Deitel, H. (2015). *Visual C# 2012 how to program* (5th Ed.). USA: Pearson Education, Inc. 2. Gaddis, T. (2016). *Starting out with visual C#* (4th Ed.). USA: Pearson Education, Inc. 3. Harwani, B. (2015). *Learning object-oriented programming in C# 5.0.* USA: Cengage Learning PTR.	

PREPARED BY:	**VERIFIED BY:**	**APPROVED BY:**
Jester Lhee I. Pandio	Brandon G. Sibbaluca, Ph.D.	Aisa Q. Hipolito, M.Ed. EL

COURSE REQUIREMENTS:	• Class participation *(Recitation, Seatwork, Quizzes, etc.)* • Major examinations • Task performance *(eLMS Activities, Laboratory Exercises, Projects, etc.)*
GRADING SYSTEM:	The following percentage distribution shall be followed: Prelims 20% Midterms 20% Pre-finals 20% Finals 40% 100% The following are the periodical grade components for this course: Class Participation 20% Task Performance 50% Major Examination 30% 100%
We'd be glad to hear from you. For questions or feedback on this course, feel free to email us through student.feedback@sti.edu	

PREPARED BY:	**VERIFIED BY:**	**APPROVED BY:**
Jester Lhee I. Pandio	Brandon G. Sibbaluca, Ph.D.	Aisa Q. Hipolito, M.Ed. EL

Course Outline

Learning Objectives (LO)		Week	LEC Hours	LAB Hours	TOPICS	Slides	I-Guide	Student Handouts	Teaching and Learning Activities	Assessment Tasks
1 2 3 4	Differentiate the features of C# from other programming languages (CO1) Analyze the components of .NET Framework (CO2) Compare the two (2) kinds of assemblies (CO2) Create a console application with basic syntax (CO4)	**1**	**2**	**3**	**.NET Framework** C# Introduction .NET Framework Structure Assembly Use Program Structure of C#				Lecture Demonstration Hands-on Activity	Seatwork 01 Quiz 1 01 Laboratory Exercise 1 01 eLMS Activity 1
5 6 7 8	Distinguish an identifier from a keyword (CO1) Declare and use constants and variables in a program (CO2) Compare value type from reference type (CO1) Convert a type of data to another data type (CO2)	**2-3**	**4**	**6**	**Data Types** Identifiers and Keywords Variables Constants Data Types Type Conversion				Lecture Demonstration Hands-on Activity	Seatwork 02 Quiz 1 02 Laboratory Exercise 1 02 eLMS Activity 1
9 10 11	Write programs that use different operators (CO2) Evaluate expressions based on order of precedence (CO1) Use the methods of the *Math* class (CO3)	**4**	**2**	**3**	**Operators and Expressions** Operators and Expressions Precedence and Associativity The *Math* Class				Lecture Demonstration Digital Learning Activity Hands-on Activity	Seatwork 03 Quiz 1 03 Task Performance 1 03 eLMS Activity 1
		5	**1**		**PRELIMINARY EXAMINATION**					Pen and Paper Test

PREPARED BY:	VERIFIED BY:	APPROVED BY:
Jester Lhee I. Pandio	Brandon G. Sibbaluca, Ph.D.	Aisa Q. Hipolito, M.Ed. EL

Learning Objectives (LO)		Week	LEC Hours	LAB Hours	TOPICS	Slides	I-Guide	Student Handouts	Teaching and Learning Activities	Assessment Tasks
12 13 14	Implement a decision using selection statements (CO2) Use the ternary operator for decision-making (CO1) Create conditions represented by logical expressions (CO2)	**6**	**2**	**3**	**Decision Making** Selection Statements The *if-else-if* Statement The *switch* Statement The Conditional Operator				Lecture Demonstration Hands-on Activity	Seatwork 04 Quiz 1 04 Laboratory Exercise 1
15 16 17	Write programs that execute statements repeatedly (CO2) Implement control using the *break* and *continue* statements (CO2) Create nested loops in a program (CO2)	**7**	**2**	**3**	**Looping** Loops The *while* Loop The *do-while* Loop The *for* Loop The *break* and *continue* Statements				Lecture Demonstration Hands-on Activity	Seatwork 05 Quiz 1 05 Laboratory Exercise 1 05 eLMS Activity 1
18 19 20	Describe the importance of arrays in programming (CO1) Declare and use one-dimensional and multidimensional arrays (CO2) Use the *ArrayList* class in a program (CO3)	**8**	**2**	**3**	**Arrays** One-Dimensional Arrays Multidimensional Arrays The *ArrayList* Class				Lecture Demonstration Digital Learning Activity Hands-on Activity	Seatwork 06 Quiz 1 06 Laboratory Exercise 1
21 22 23	Create and use strings in programs (CO4) Perform operations on strings using various methods (CO4) Use the *StringBuilder* class in a program (CO3)	**9**	**2**	**3**	**Strings** Strings The *StringBuilder* Class				Lecture Demonstration Hands-on Activity	Seatwork 07 Quiz 1 07 Task Performance 1 07 eLMS Activity 1
		10	**1**		**MIDTERM EXAMINATION**					Pen and Paper Test

PREPARED BY:	VERIFIED BY:	APPROVED BY:
Jester Lhee I. Pandio	Brandon G. Sibbaluca, Ph.D.	Aisa Q. Hipolito, M.Ed. EL

Learning Objectives (LO)		Week	LEC Hours	LAB Hours	TOPICS	Slides	I-Guide	Student Handouts	Teaching and Learning Activities	Assessment Tasks
24 25 26	Distinguish method from structure (CO1) Implement method overloading (CO3) Create programs that define methods and their parameters (CO4)	**11**	**2**	**3**	**Methods and Structures** Methods Method Overloading Structures				Lecture Demonstration Hands-on Activity	Seatwork 08 Quiz 1 08 Laboratory Exercise 1 08 eLMS Activity 1
27 28 29 30 31	Use classes and the *new* operator (CO1) Create programs that access public members (CO3) Apply encapsulation in programs (CO3) Use the *this* keyword (CO1) Implement the constructor overloading (CO3)	**12-13**	**4**	**6**	**Classes and Objects** Class Encapsulation The *this* Keyword Constructors Namespace				Lecture Demonstration Digital Learning Activity Hands-on Activity	Seatwork 09 Quiz 1 09 Laboratory Exercise 1 09 Task Performance 1
		14	**1**		**PRE-FINAL EXAMINATION**					Pen and Paper Test
32 33 34 35 36	Use inheritance and protected members (CO1) Invoke base class constructors (CO3) Apply method overriding (CO3) Use the abstract classes and abstract methods (CO2) Implement polymorphism in a program (CO3)	**15-16**	**4**	**6**	**Inheritance and Polymorphism** Inheritance Method Overriding Abstract Classes Polymorphism				Lecture Demonstration Digital Learning Activity Hands-on Activity	Seatwork 10 Quiz 1 10 Laboratory Exercise 1 10 Laboratory Exercise 2 10 eLMS Activity 1
37 38	Construct programs that invoke interface members at the object level (CO4) Implement interfaces in a class (CO3)	**17**	**2**	**3**	**Interfaces** Invoking Interface Members at the Object Level Implementing Multiple Interfaces				Lecture Demonstration Hands-on Activity	Seatwork 11 Quiz 1 11 Task Performance 1

PREPARED BY:	VERIFIED BY:	APPROVED BY:
Jester Lhee I. Pandio	Brandon G. Sibbaluca, Ph.D.	Aisa Q. Hipolito, M.Ed. EL

Learning Objectives (LO)		Week	LEC Hours	LAB Hours	TOPICS	Slides	I-Guide	Student Handouts	Teaching and Learning Activities	Assessment Tasks
39	Declare the properties on an interface (CO2)				Implementing Interface Properties					
		18	**1**		**FINAL EXAMINATION**					Pen And Paper Test

PREPARED BY:	VERIFIED BY:	APPROVED BY:
Jester Lhee I. Pandio	Brandon G. Sibbaluca, Ph.D.	Aisa Q. Hipolito, M.Ed. EL