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's response regardless of circumstances is consistently virtuous. **Critical thinker (IO2):** An STIer is a critical thinker. An STIer discerns through open-minded analysis and challenges one's beliefs to further improve creative ideas. **Communicator (IO3):** An STIer communicates to understand and be understood. An STIer recognizes the emotion behind the information read or heard and expresses his/her own emotion when giving information, may it be verbal or written. **Change adept (IO4):** An STIer is change adept. An STIer utilizes such ability to manage when a need for transformation or transition arises to uplift his/her capabilities and build the capacity of the community where he/she belongs.

SERIAL NUMBER: IT1712	**COURSE TITLE:** COMPUTER PROGRAMMING 2	**CREDIT:** 2 lec, 1 lab (2 hours lecture and 3 hours laboratory per week)
COURSE DESCRIPTION:	This course is a continuation of Computer Programming 1. The emphasis is to train students to design, implement, test, and debug programs intended to solve computing problems using standard libraries.	
PREREQUISITE:	COMPUTER PROGRAMMING 1	
COURSE OUTCOMES:	After successful completion of this course, the student should be able to: (ITPO01, ITPO02 – PI1, ITPO04 – PI2, ITPO06 – PI3, ITPO08 – PI4, ITPO08 – PI5) CO1. analyze the identified problem specification to come up with possible and best solutions; CO2. design, implement, test, and debug a program, based on a given specification that uses standard libraries in the programming language; CO3. apply the latest recommended technologies to programs using the object-oriented approach; and CO4. assess and recommend tools and techniques to programs created by other members of a project's team.	
MANDATED BOOK:	None	
REFERENCES:	1. Baesens, B., Backiel, A., & Broucke, S. (2015). *Beginning java programming: The object-oriented approach.* Indiana: John Wiley & Sons, Inc. 2. Savitch, W. (2014). *Java: An introduction to problem solving and programming, 7th edition.* California: Pearson Education, Inc. 3. Farrell, J. (2014). *Java programming, 7th edition.* Boston: Course Technology, Cengage Learning	
COURSE REQUIREMENTS:	• Class Participation (Recitation, Seatwork, Quizzes) • Major Examinations	

PREPARED BY: Beronika A. Peña	**VERIFIED BY:** Brandon G. Sibbaluca, PhD	**APPROVED BY:** Aisa Q. Hipolito, MEd. EL

	- Task Performance (Laboratory Exercises, eLMS Activities)
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: Beronika A. Peña	**VERIFIED BY:** Brandon G. Sibbaluca, PhD	**APPROVED BY:** Aisa Q. Hipolito, MEd. EL

Course Outline

Learning Objectives (LO)		Week	LEC Hours	LAB Hours	TOPICS	Slides	I-Guide	Student Handout	Teaching and Learning Activities	Assessment Tasks
I 2	Determine the benefits of OOP approach (CO1) Describe the basic techniques of program design in object-oriented programming (CO1)	**1-2**	**4**	**6**	**Object-Oriented Programming Concepts** Object Class Inheritance Interface Package				Java Game (01 Activity 1) Lecture Demonstration	01 Laboratory Exercise 1 01 eLMS Quiz 1
4 5 6	Link the relationship between a class and an object of a class (CO1) Apply the concepts of information hiding and encapsulation in a program (CO2, CO3) Use variables of a class type to name objects (CO2)	**3**	**2**	**3**	**Classes and Methods** Class and Method Definitions Information Hiding and Encapsulation Objects and References				Java Game (02 Activity 1) Lecture Demonstration	02 eLMS Activity 1 02 Laboratory Exercise 1
7 8 9 10 11	Use constructor, static variables, and methods in a class (CO2) Construct overloaded methods and statements using methods from the Math class (CO2) Determine the role of inheritance to produce interfaces within application programs (CO1) Describe polymorphism in general (CO1) Define interfaces to specify methods (CO2)	**4**	**2**	**3**	**Objects and Methods** Constructors Static Variables and Static Methods **Inheritance, Polymorphism, and Interfaces** Inheritance Polymorphism Interface and Abstract Classes				Java Game (03 Activity 1) Lecture Demonstration Java Game (04 Activity 1)	03 eLMS Activity 1 04 eLMS Quiz 1 04 Task Performance 1

PREPARED BY: Beronika A. Peña	**VERIFIED BY:** Brandon G. Sibbaluca, PhD	**APPROVED BY:** Aisa Q. Hipolito, MEd. EL

Learning Objectives (LO)		Week	LEC Hours	LAB Hours	TOPICS	Slides	I-Guide	Student Handout	Teaching and Learning Activities	Assessment Tasks
		5			**PRELIMINARY EXAMINATION**					Pen and Paper Test
12 13 14	Justify the importance of exception handling (CO1) Write programs that handle exceptions correctly (CO2, CO3, CO4) Create customized *Exception* classes (CO2)	**6-7**	**4**	**6**	**Exception Handling** Basic Exception Handling *try*, *catch*, and *finally* User-Defined Exceptions				PPT Game (05 Activity 1) Lecture Demonstration Java Game (05 Activity 2)	05 Laboratory Exercise 1 05 eLMS Quiz 1 05 Laboratory Exercise 2
15 16 17 18 19	Compare text files and binary files (CO1) Use the *Path* and *File* classes in programs (CO2) Describe file organization, streams, and buffers (CO1) Use the IO classes in writing to and reading from a file (CO2) Create and use sequential data files (CO2, CO3)	**8-9**	**4**	**6**	**File Input and Output** Computer Files The *Path* and *File* Classes File Organization, Streams, and Buffers The IO Classes Sequential Data Files				Java Game (06 Activity 1) Lecture Demonstration Java Game (06 Activity 2)	06 Laboratory Exercise 1 06 eLMS Quiz 1 06 Task Performance 1
		10			**MIDTERM EXAMINATION**					Pen and Paper Test
20 21	Use enumerated types Construct nested classes	**11**	**2**	**3**	**Enumerations and Nested Classes** Enumerations Nested Classes				Lecture Demonstration	07 Laboratory Exercise 1
22	Search, parse, and replace strings using regular expressions	**12-13**	**4**	**6**	**Regular Expressions** Fundamentals Repetition Operators Pattern and Matcher RegEx Operations				Lecture Demonstration	08 Laboratory Exercise 1 08 eLMS Quiz 1 08 Task Performance 1
		14			**PRE-FINAL EXAMINATION**					Pen and Paper Test

PREPARED BY: Beronika A. Peña	**VERIFIED BY:** Brandon G. Sibbaluca, PhD	**APPROVED BY:** Aisa Q. Hipolito, MEd. EL

Learning Objectives (LO)		Week	LEC Hours	LAB Hours	TOPICS	Slides	I-Guide	Student Handout	Teaching and Learning Activities	Assessment Tasks
23	Create and control threads	**15**	**2**	**3**	**Concurrency** Fundamentals Thread Methods				Lecture Demonstration	09 Laboratory Exercise 1
24 25	Create and use a generic class Use wildcard arguments	**16-17**	**4**	**6**	**Generics** Fundamentals Wildcard Arguments				Lecture Demonstration	10 Laboratory Exercise 1 10 eLMS Quiz 1 10 Task Performance 1
		18			**FINAL EXAMINATION**					Pen and Paper Test

PREPARED BY: Beronika A. Peña	**VERIFIED BY:** Brandon G. Sibbaluca, PhD	**APPROVED BY:** Aisa Q. Hipolito, MEd. EL