## College of Information Technology and Computing
Department of Information Technology

**SYLLABUS**
Course Title: **Object Oriented Programming**
Course Code: **IT214**
Credits: 3 units (2 hours Lecture, 3 hours Laboratory)

**USTP Vision**

A nationally-recognized Science and Technology (S&T) university providing the vital link between education and the economy

**USTP Mission**

Bring the world of work (industry) into the actual higher education and training of the students;
Offer entrepreneurs of the opportunity to maximize their business potentials through a gamut of services from product conceptualization to commercialization;
Contribute significantly to the national development goals of food security and energy sufficiency through technology solutions.

Semester/Year: **1st Semester SY 2025-2026**
Class Schedule:
- **IT2R1- T 1:00-4:00 | Th 8:00 – 10:00 A**
- **IT2R2- W 1:00 -4:00 | Th 12:00 – 2:00 P**
- **IT2R3- W 1:00-4:00 | Th 10:00-12:00N**
- **IT2R4- M 1:00-4:00 | T 3:00 – 5:00 P**
- **IT2R5- W 10:00 – 1:00 | Th 8:00 – 10:00 A**
- **IT2R6- Th 1:00-4:00 | W 8:00-10:00 A**
- **IT2R7- W 7:00-10:00 | Th 10:00-12:00 N**
- **IT2R8- F 1:00 – 4:00 | W 10:00-12:00 N**
- **IT2R9- T 5:00 – 7:00 | W 5:00 – 8:00 P**

Bldg./Rm. No.
- Lec.: **OC** Lab: **ICT Building 9**

Prerequisite(s): **IT121 , IT122**
Co-requisite(s):

Instructor: **Jocelyn L. Garrido**
Email: **jocelyn.garrido@ustp.edu.ph**
Mobile No.: **+639656510966**

Instructor: **Jhon Harvey Babia**
Email: **jhonharvey.babia@ustp.edu.ph**

Instructor: **Charlane Vallar, MIT**

Consultation Schedule: **Friday 1:00 – 5:00 PM**
Bldg.Rm. No.: **09 – 309**
Office Phone No./Local: **(088) 856 1739 local 1153**

**USTP Core Values:**

A. **Unselfish Dedication** – Selfless commitment and complete fidelity towards a course of action or goal.
B. **Social Responsiveness** – Ethical/moral responsibility leading to corrective action on social issues and contributions for the betterment of the environment and the community's quality of life.
C. **Transformational Leadership** – Leading through inspiration and by example to foster positive change with the end goal of developing followers into leaders.
D. **Prudence –** Self-governance leading to circumspection and good judgment in the management of affairs and use of resources.

**Program Educational Objectives:**

**PEO1:** Graduates are proficient in the IT field and able to engage constantly in technological and professional advancement by pursuing a higher academic level and practicing quality

Email: **charlane.vallar@ustp.edu.ph**

## I. Course Description:

This course introduces students to Object-Oriented Programming (OOP), focusing on fundamental concepts such as classes, objects, inheritance, polymorphism, encapsulation, abstraction, and exception handling. Students will learn to design, implement, and debug OOP based applications such as Java using GUI development, file handling, and database connectivity.

## II. Course Outcomes:

| Course Outcomes (CO) | Program Outcomes (PO) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| **CO1:** Apply OOP concepts using Java to design and implement modular and efficient solutions to computing problems. | E | I | | | D | I | E | | | | | | | | E |
| **CO2:** Collaboratively design, develop, and present an OOP-based system project that integrates GUI, file handling, and database connectivity to meet specified requirements and industry standard. | E | D | I | D | D | D | | D | I | D | I | E | E | E | D |

## III. Course Outline:

| Allotted Time | Course Outcomes (CO) | Intended Learning Outcomes (ILO) | Topic/s | Suggested Readings | Teaching-Learning Activities | Assessment Tasks/Tools | Grading Criteria | Remarks |
|---|---|---|---|---|---|---|---|---|
| 5 hrs<br>Week 1<br>Aug 11-16 | CO1 | • Explain the role of Java in system development | • Course Orientation (Class Policies & requirements)<br>• **Introduction to Java and OOP** | • Student Handbook<br>Course Syllabus | - Orientation<br>- Concept Discussions and Demo | - Short Quiz on Java basics. | - Quiz Score | |

improvement in their career and personal lives.

**PEO2:** Graduates are competent in generating new ideas and innovations in Information Technology with more emphasis on technopreneurship, management, IT solutions and the likes through research collaborations.

**PEO3:** Graduates are practicing professionals in the field of Information Technology who can contribute significantly to human development, socio-economic transformation, and patriotic initiatives.

**Program Outcomes:**

**a:** Identify, select and apply appropriate knowledge of computing science and mathematics in solving computing problems.

**b:** Understand, apply and integrate best practices and standards in solving computing problems by evaluating their applications

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | • Write and run a simple Java program | - Java program structure & syntax<br>- OOP Principle Overview<br>- Java platform, installation and environment (JDK, JRE, IDE setup) | • Reference Textbook<br>• Online references | - IDE setup | | | |
| 5 hrs<br>Week 2<br>Aug 18-23 | CO1 | • Use variables and operators effectively.<br>• Differentiate between primitive and object references | • **Data Types, Variables, and Operators**<br>- Primitive & reference data types.<br>- Operators (arithmetic, relational, logical)<br>- Type casting | • Reference Textbook<br>• Online references | • Concept Discussions and Demo<br>• Guided Coding<br>• Coding Activity | - Coding Exercises Submission<br>- Assignment | Rubric | |
| 5 hrs<br>Week 3<br>Aug 25-30 | CO1 | • Implement decision-making and looping constructs in Java programs | • **Control Structures**<br>- Conditional statements `if`, `if-else`, `nested if`, `switch`)<br>- Iterators and Loops (`for`, `while`, `do-while`) | • Reference Textbook<br>• Online references | • Lecture and Demo<br>• Problem Solving task | - Coding Exercise on Menu-Driven application.<br>- Quiz | Rubrics<br>Quiz Score | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **c:** Work collaboratively among members of the team to analyze complex problems by applying analytical and quantitative reasoning; and define the computing requirements appropriate to its solution.<br><br>**d:** Communicate effectively with users to identify their needs and apply critical and creative thinking skills to do analysis and take them into account in the selection, creation, evaluation and administration of computer-based systems.<br><br>**e:** Creatively design, implement and evaluate using different computer-based systems, processes, components, or programs to meet desired needs and requirements under various constraints<br><br>**f:** Properly integrate IT-based solutions using various methods, policies and processes into the user environment effectively.<br><br>**g:** Apply and demonstrate knowledge through the use of current techniques, skills, tools, methods, theory and practices necessary for the IT profession with diversity and multicultural competencies to promote equity | 5 hrs.<br>Week 4<br>Sep 1-6 | CO1 | • Modularize code using reusable methods<br>• Apply method overloading | • **Methods and Parameter Passing**<br>- Defining & invoking methods<br>- Method overloading<br>- Parameter passing | • Reference Textbook<br>• Online references | • Lecture<br>• Hands-on coding | - Long Quiz (Prelim) | Quiz Score | |
| | 5 hrs.<br>Week 5<br>Sep 8 -13 | CO1 | • Define and use classes and objects<br>• Apply constructors and the `this` keyword | • **Classes and Objects**<br>- Defining classes<br>- Creating objects<br>- Fields, constructors, `this` keyword | • Reference Textbook<br>• Online references | • Lecture and Demo<br>• Problem Solving task | - Mini class modeling Exercise | Rubric | |
| | 5 hrs<br>Week 6<br>Sep 15 -20 | CO1 | • Apply encapsulation and data hiding using access modifiers | • **Encapsulation and Access Modifiers**<br>- Getters & setters<br>- `private`, `public`, `protected`, default<br>- Data hiding | • Reference Textbook<br>• Online references | • Lecture<br>• Coding workshop | - Code review and feedback | | |
| | 5 hrs.<br>Week 7<br>Sep 22-27 | CO1 | • Create class hierarchies<br>• Apply method overriding and polymorphism | • **Inheritance and Polymorphism**<br>- `extends` keyword | • Reference Textbook | • Lecture and Demo | - Coding Quiz | Quiz Score | |

and social justice in the community.

**h:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings by developing and contributing positively to the accomplishment of team goals through collaborative process, developing and practicing effective interpersonal skills

**i:** Assist in the creation of an effective IT Project Plan by evaluates as individual and team's values and sense of responsibility through participation in a range of learning contexts.

**j:** Communicate effectively in English (and as much as possible using local language and Filipino) with the computing community and with society at large about complex computing activities through interviewing, logical and ethical writing, presentations, and clear instructions

**k:** Able to work collaboratively and respectfully as members and leaders of diverse teams and communities in analyzing, understanding, and assessing

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | - Method overriding<br>- Upcasting & downcasting | • Online references | • Coding Lab Exercise | | | |
| 5 hrs.<br>Week 8<br>Sep 28-Oct 1 | CO1 | • Design programs using abstraction via abstract classes and interfaces | • **Abstract Classes and Interfaces**<br>- Abstract classes & methods<br>- Interfaces & multiple inheritance | • Reference Textbook<br>• Online references | • Lecture<br>• Demo – interface implementation | - Lab Exercise | Rubrics | |
| Week 9<br>Oct 2-8 | **MIDTERM EXAMINATION** | | | | | | | |
| 10 hrs.<br>Week 10-11<br>Oct 9-18 | CO1 | • Use arrays and collection classes to store and process data<br>• Implement robust error handling with try-catch-finally<br>• Create custom exceptions | • **Arrays, Collections Framework and Exception Handling**<br>- Single & multi-dimensional arrays<br>- ArrayList, LinkedList, HashMap<br>- try-catch-finally<br>- Throwing exceptions | • Reference Textbook<br>• Online references | • Lecture<br>• Coding Demo<br>• Code walk-through | - Coding Quiz | Quiz Result | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | - Custom exceptions | | | | | |
| 5 hrs. Week 12 Oct 20-25 | CO1 | • Perform file read/write operations<br>• Apply object serialization | • **File I/O and Serialization**<br>- Reading & writing text files<br>- Object serialization/ deserialization | • Reference Textbook<br>• Online references | • Lecture and Demo<br>• Code Review | - Coding Quiz | Quiz Result | |
| 5 hrs. Week 13 Oct 27 - 31 | CO1 & CO2 | • Create basic GUI applications<br>• Handle events<br>• Apply layout managers | • **GUI Development with JavaFX / Swing**<br>- Basic UI components<br>- Event handling<br>- Layout managers | • Reference Textbook<br>• Online references | • Lecture<br>• Demo on Building GUI | - GUI Mini-project | Rubrics | |
| 5 hrs Week 14 Nov 3-8 | CO1 & CO2 | • Connect to a database<br>• Perform CRUD operations using JDBC | • **Database Connectivity (JDBC)**<br>- Connecting to MySQL<br>- CRUD operations<br>- Prepared statements | • Reference Textbook<br>Online references | • Lecture and Demo<br>• CRUD Application Demo | - Database Linked Application Activity (Semi-Final) | Practical Quiz | |
| 5 hrs Week 15 Nov 10-15 | CO2 | • Create UML diagrams | • **System Design Principles** | | | | Approved Proposal | |

societal issues and act responsibly in making design and implement decisions considering the result of the research relevant to the local and global impact on computing information technology on the Filipino culture, individuals, organizations, and society.

**l :** Understand professional, ethical, legal, security and social issues and responsibilities in the utilization of information technology.

**m :** Apply professional, ethical, legal, security and social issues and responsibilities in the utilization of information technology. Understand, assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice

**n :** Participate in generation of new knowledge or in research and development projects aligned to local and national development agenda or goals

**o :** Graduates are able to apply and demonstrate sufficient expertise in the field of Information Technology with the

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| end view of contributing to the local and national economy. | | | • Apply SDLC and MVC principles in planning | - UML diagrams (class, sequence)<br>- Software development life cycle (SDLC)<br>- Design patterns (MVC) | • Reference Textbook<br><br>Online references | • Group project planning and diagram creation | - Project Proposal Submission | | |
| | 10 hrs.<br>Week 16 – 17<br>Nov 17-Dec 4 | CO2 | • Implement modular, integrated OOP system with GUI and database<br>• Test, debug, document, and present a complete OOP-based Java system | • System Development Implementation and Presentation<br>- Integrating OOP, GUI, database<br>- Modular coding practices<br>- Testing & debugging<br>- Documentation<br>- System demo | • Reference Textbook<br><br>Online references | • Group coding sessions.<br><br>• Group project completion. | - Project Presentation | Project and Presentation Rubrics | |
| | Week 18<br>Dec 5-13 | **FINAL EXAMINATION** | | | | | | | |

**IV. Course Requirements:**

1. Class standing (attendance, participation, etc.) policy:
   (a) Expected classroom behavior (may want to develop this with the students, e.g., What guidelines m are appropriate for behavior and participation in a large class
   - Students must come to class on time.
   - Strict observance of deadlines.
   - Class participation is encouraged.
   - Observe proper courtesy.

(b) Ground Rules for participation in discussions or activities.

- Only one student may talk at a time.
- Must follow instructions for every activity given.
- For group activity, each member must participate accordingly.

**2.** Course Readings/Materials:

(a) Titles, authors, and editions of textbooks and other materials, required and recommended

1. Java for Programmers, P.J.Deitel and H.M.Deitel, PEA
2. Java: How to Program , P.J.Deitel and H.M.Deitel,PHI
3. Object Oriented Programming through Java, P.Radha Krishna, Universities Press
4. NetBeans IDE 8 Cookbook. David Salter and Rhawi Dantas. Packt Publishing

(b) Supplies needed (calculators, software, workbooks, disks, CDs, lab supplies, art supplies, etc.)

- Javascript/Type Scripting Software
- C/C++ Programming Software
- Python Programming Software IDE
- Java Programming (JCreator, NetBeans)
- MySQL/ MariaDB/ SQLLite

(c) URLs for online resources

- Java Complete Reference 9th Edition by: Herbert Schildt
  https://www.sietk.org/downloads/javabook.pdf
- Java OOP
  https://www.w3schools.com/java/java_oop.asp

**3.** Assignments, Assessment, and Evaluation

(a) Policy concerning homework (grading, posting, late policy, etc.)

Students may share ideas as they work on their assignments but the submitted assignments must be their own work.

(b) Policy concerning make-up exams

No special examination is given unless a student has valid reasons stipulated in the Student Handbook Article 3: Excused Absences.

(c) Policy concerning late assignments/requirements

- Assignments: no assignment for a particular date, will have a grade of zero (0).
- Projects: late submission of projects will have a corresponding consequence. There will be a deduction of points for every day that the project submission will be late.

(d) Preliminary information on term papers or projects, with due dates
- Projects for midterm and finals are given ahead of time along with its corresponding due dates, rubrics, and other requirements for the completion of the projects.
- Non-submission of projects does not mean you

(e) List of assignments that will impact the final grade and % weight given each
- Portfolio: grade will be part of the PIT.

(f) Description in detail of grading processes and criteria (how many quizzes, tests, papers; weighting of each; amount of homework, etc.) or the GRADING POLICY

**Grading System**

| Lecture Grade (67%) | |
|---|---|
| **Performance Item/Criteria** | ***%*** |
| Class Performance Item | 10% |

| | |
|---|---|
| Quizzes (All quizzes, prelim and pre-final exams) | 40% |
| Major Exams (i.e, Midterm and Final Exams) | 30% |
| Performance Innovative Task / Project | 20% |
| **TOTAL** | **100%** |

| **Laboratory Grade (33%)** | |
|---|---|
| **Performance Item/Criteria** | ***%*** |
| Laboratory Exercises/Reports | 30% |
| Laboratory Major Exam | 40% |
| Hands on Exercises | 30% |
| **TOTAL** | **100%** |

**Term/Periodic Grade = 67% Lecture Grade + 33% Laboratory Grade**

**Options:**

**FINAL GRADE (FG) = 1/3 Midterm Grade (MTG)+ 2/3 Final Term Grade (FTG)**

**FINAL GRADE (FG) = 1/2 Midterm Grade (MTG)+ 1/2 Final Term Grade (FTG)**

**(Passing Percentage is 70%)**

Ex. In a 10-item quiz, obtaining 7 points would be equivalent to a passing score.

**Rubrics:**

**A. Laboratory Coding Exercise**

| Criteria | Excellent (5) | Proficient (4) | Satisfactory (3) | Needs Improvement (2) | Poor (1) |
|---|---|---|---|---|---|
| **Code Functionality (50%)** | Runs perfectly with all required features implemented; meets all specifications. | Runs with minor issues; most features work. | Runs but some features missing or not working. | Runs with major errors; limited functionality. | Does not run or produces wrong results. |
| **Application of OOP Concepts (35%)** | Correct and effective use of OOP principles (classes, objects, inheritance, etc.) | Mostly correct use of OOP principles with small errors. | Some OOP concepts used but with multiple mistakes. | Minimal use of OOP; mostly procedural code. | No OOP concepts applied. |
| **Code Quality & Style (15%)** | Well-structured, readable, and follows naming conventions. | Mostly well-structured; minor style issues. | Readable but inconsistent formatting. | Poorly formatted; hard to read. | Very messy and unreadable code. |

**B. Final System Development Project**

| Criteria | Excellent (5) | Proficient(4) | Satisfactory (3) | Needs Improvement (2) | Poor (1) | Weight |
|---|---|---|---|---|---|---|
| **System Functionality** | All CRUDS features work perfectly; meets all requirements; handles edge cases gracefully. | All CRUDS features work with minor bugs. | Most CRUDS features work but some partially implemented. | Only 2–3 CRUDS functions work; several major bugs. | System does not work or is incomplete. | **50%** |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Application of OOP Concepts** | Correct and efficient use of OOP principles: classes, objects, encapsulation, inheritance, polymorphism, abstraction. | OOP principles applied mostly correctly; minor design flaws. | Some OOP principles applied but with significant issues. | Minimal use of OOP; mostly procedural approach. | No OOP concepts applied. | **25%** |
| **Code Quality & Structure** | Code is well-organized, modular, readable, and follows naming conventions; reusable methods/classes used effectively. | Mostly organized; minor naming or structure issues. | Understandable but lacks modularity and consistency. | Poorly structured; messy or repetitive code. | Very disorganized; unreadable code. | **15%** |
| **User Interface & Usability** | Clean, intuitive interface; easy to navigate; user-friendly error handling. | Mostly user-friendly; minor design or navigation issues. | Functional but not visually appealing; some navigation issues. | Confusing or poorly designed interface. | No clear interface; difficult to use. | **10%** |

**C. Oral Presentation and Defense**

| Criteria | Excellent (5) | Good (4) | Satisfactory (3) | Needs Improvement (2) | Poor (1) | Weight |
|---|---|---|---|---|---|---|
| **Clarity & Organization** | Presentation is well-structured, logical flow, clear introduction, body, and conclusion. | Mostly organized with minor lapses in flow. | Some organization but transitions are rough. | Disorganized and difficult to follow. | No clear structure; confusing. | **20%** |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Content Knowledge** | Demonstrates deep understanding of the topic; answers questions confidently with accurate information. | Good understanding; minor inaccuracies or hesitations in answers. | General understanding; some inaccurate or vague answers. | Limited understanding; struggles with answers. | Unable to explain or answer questions. | **30%** |
| **Technical Accuracy** | All technical details are correct and well-explained using appropriate terms. | Most technical details correct with minor errors. | Several errors in technical explanation. | Many technical inaccuracies. | Mostly incorrect technical information. | **20%** |
| **Delivery & Communication Skills** | Speaks clearly, maintains good pacing, eye contact, and audience engagement; minimal filler words. | Mostly clear delivery; minor pacing or engagement issues. | Understandable but often too fast, too slow, or monotone. | Hard to understand; poor pacing; low engagement. | Mumbled, unclear, or inaudible. | **15%** |
| **Use of Visual Aids** | Visuals (slides/demo) are clear, professional, and enhance understanding. | Visuals mostly clear; minor design or clarity issues. | Visuals somewhat helpful but cluttered or hard to read. | Poorly designed visuals; distract from message. | No visuals or visuals irrelevant. | **15%** |

*Disclaimer:*
*Every attempt is made to provide a complete syllabus that provides an accurate overview of the subject. However, circumstances and events make it necessary for the instructor to modify the syllabus during the semester. This may depend, in part, on the progress, needs, and experiences of the student.*

| **Prepared by:** | **Recommending Approval:** | **Approved by:** |
|---|---|---|
| **JOCELYN L. GARRIDO** | **DR. LOVE JHOYE M. RABOY** | **DR. JUNAR A. LANDICHO** |
| Faculty, Dept. of Information Technology | Chairperson, Dept. of Information Technology | Dean, CITC |