

Daily Induction Plan (DIP) Guidelines and Reference

DIP Template Structure

Header Information

```
# Daily Induction Plan

**Lecturer:** Muhammad Aiman Syahmi Bin Haris
**Subject:** CP115
**Week:** [Week Number]
```

Course Learning Outcomes (CLOs)

Standard CLOs for CP115:

- **CLO1:** Explain basic computer programming concepts
- **CLO2:** Solve simple problems using algorithms and a programming language
- **CLO3:** Demonstrate programming skills in solving simple problems

Class Schedules and CLO Usage

C01 Class Schedule:

- **Tuesday:** Lab 1 (11:15 AM - 1:15 PM) + Tutorial 1 (7:15 AM - 8:15 AM)
- **Wednesday:** Tutorial 2 (7:15 AM - 8:15 AM)
- **Thursday:** Lecture 2 (7:15 AM - 8:15 AM)

C02 Class Schedule:

- **Monday:** Lab 1 (11:15 AM - 1:15 PM)
- **Tuesday:** Tutorial 1 (2:15 PM - 3:15 PM)
- **Thursday:** Tutorial 2 (10:15 AM - 11:15 AM)

Combined Lectures (Both Classes):

- **Wednesday:** Lecture 1 (8:15 AM - 9:15 AM) - C01 & C02 together
- **Thursday:** Lecture 2 (7:15 AM - 8:15 AM) - C01 & C02 together

CLO Usage Guidelines:

Class Type	CLO to Use	Notes
PRAKTIKUM (Labs)	CLO2 or CLO3	Always practical application
LECTURE	CLO1	Always conceptual understanding
TUTORIAL	CLO1, CLO2, or both	Depends on content mix

Table Structure

```
| # | Day/Time | Class | Mode | Topic | Learning Outcome | CLO | Reflection |
|---|-----|-----|-----|-----|-----|-----|-----|
```

Space-Saving Guidelines

- **Combined Lectures:** Lectures are shared between C01 and C02 classes, so combine them into single entries in the DIP table to save space
- **Example:** Instead of separate rows for C01 Lecture 1 and C02 Lecture 1, use one row with "C01 & C02" in the Class column

Content Sources

Refer to these documents for content:

1. **Course Schedule:** docs/course/schedule.md - Main course content and topics
2. **C01 Schedule:** docs/course/c01-schedule.csv - C01 class dates
3. **C02 Schedule:** docs/course/class-schedule.csv - C02 class dates

Signature Section Format

```
---

<br><br>

**Prepared By:**

<br><br><br>

Muhammad Aiman Syahmi Bin Haris

<br><br>

**Verified By:**

<br><br><br>

Zulkarnaen Bin Saridi
Head of Computer Science Unit
```

Weekly Content Mapping

Content Format Guidelines

- **Topics Column:** Only put topic names (e.g., "Topic 1: Introduction to Programming", "Topic 2: Problem Analysis")
- **Learning Outcome Column:** Copy exact text from schedule.md based on weeks (e.g., "(a) Identify programming concepts", "(c) Explain five (5) steps in problem solving")
- **Reflection Column:** Must be suitable and relevant to the specific topic and learning outcome being taught

Week 1 Content Mapping

From schedule.md Week 1:

- **Lecture:** Course briefing + Topic 1: Introduction to Programming - (a) Identify programming concepts
- **Tutorial:** Course briefing
- **Praktikum:** Course briefing

Week 2 Content Mapping

From schedule.md Week 2:

- **Lecture:** Topic 1: Introduction to Programming - (b) List five (5) steps in problem solving + Topic 2: Problem Analysis - (a) Define the Input-Process-Output (IPO) model
- **Tutorial:** Topic 1: Introduction to Programming - (c) Explain five (5) steps in problem solving
- **Praktikum:** Topic 1: Introduction to Programming - (d) Demonstrate use of design software and coding platforms

(Continue mapping based on schedule.md)

Reflection Writing Guidelines

Topic-Specific Reflections:

Reflections must be relevant to the specific topic and learning outcome. Match reflection content to what students are actually learning.

Tone and Variety:

- Mix positive and realistic reflections
- Include some challenges students face
- Use generic, professional language
- Make reflections specific to the topic and learning outcome

Examples by Topic/Learning Outcome:

Course Briefing:

- **Good:** "Students understand the course structure, assessment methods, and expectations clearly"
- **Realistic:** "Some students need clarification on assessment weightages and submission procedures"
- **Mixed:** "Most students grasp the course overview, though some require additional explanation of practical requirements"

Topic 1 - (a) Identify programming concepts:

- **Good:** "Students can identify basic programming concepts and terminology effectively"
- **Realistic:** "Students have difficulty distinguishing between different programming paradigms initially"
- **Mixed:** "Most students understand fundamental concepts, though some struggle with abstract programming terminology"

Topic 1 - (c) Explain five (5) steps in problem solving:

- **Good:** "Students can systematically apply the five problem-solving steps to given scenarios"
- **Realistic:** "Students find it challenging to break down complex problems into manageable steps"
- **Mixed:** "Most students understand the problem-solving methodology, though some need practice applying it consistently"

Topic 1 - (d) Demonstrate use of design software and coding platforms:

- **Good:** "Students successfully navigate and utilize programming environments and design tools"
- **Realistic:** "Students experience initial difficulties with software installation and interface navigation"
- **Mixed:** "Most students adapt to the programming tools well, though some need additional support with advanced features"

Topic 4 - (d) Construct Python programs based on components:

- **Good:** "Students can construct basic Python programs using proper components and syntax structure"

- **Realistic:** "Students have difficulty remembering all Python syntax rules and proper indentation initially"
- **Mixed:** "Most students understand Python components well, though some need practice with data type conversions"

Topic 5 - (d) Apply the IPO model, pseudocode and flowchart (sequence):

- **Good:** "Students successfully apply the IPO model and can create effective sequential flowcharts for problem-solving"
- **Realistic:** "Students struggle with translating complex problems into clear sequential flowchart format"
- **Mixed:** "Most students understand sequential logic, though some struggle with translating problems into flowchart format"

Topic 6 - (c) Identify relational and logical operators in selection structures:

- **Good:** "Students can explain the concepts of selection structures and apply logical operators in different programming scenarios"
- **Realistic:** "Students can identify relational operators but find logical operators challenging to distinguish initially"
- **Mixed:** "Most students understand basic selection concepts, though some struggle with complex logical combinations"

File Naming Convention

- `week-[number]-dip.md`
- Example: `week-1-dip.md` , `week-2-dip.md`

Navigation Integration

Update `.vitepress/config.mjs` to add new DIP weeks:

```
{
  text: 'Daily Induction Plans',
  items: [
    { text: 'Week 1 DIP', link: '/course/DIP/week-1-dip' },
    { text: 'Week 2 DIP', link: '/course/DIP/week-2-dip' },
    // Add more weeks as needed
  ]
}
```