

# Assignment 06: Assignment and practice of Prompt Engineering to craft effective prompts.

## Assignment Tasks:

### Task 1: Prompt Categorization

Categorize the following prompts into types (Instructional, Conversational, Visual, etc.) and explain your reasoning:

1. Generate a logo for a tech startup using neon colors:



#### Prompt:

“Generate a logo for a tech startup using neon colors.”

#### Type:

Visual / Creative Generation

#### Why it fits:

Imperative verb “Generate”; asks for an image asset with a style constraint (neon).

#### Typical output:

Logo concepts or image files; maybe color palette & style notes.

2. Explain blockchain to a 5-year-old:

One day, some kids in the park started trading candies.

But sometimes kids would **forget** who gave what, or someone would **cheat** and say, “*Hey! You owe me 5 candies!*” when it wasn’t true.

So, they made a rule:

- Every time someone gives or trades a candy, they **write it down on a special page**.
- Then, they **show the page to all the kids**.
- All the kids copy the same thing in their own books. Now, if one kid tries to lie later, everyone else's books will prove the truth.

This big shared candy-book is just like a **blockchain** — it keeps everyone honest and makes sure candy trades are fair.

**Prompt:**

“Explain blockchain to a 5-year-old.”

**Type:**

Instructional / Explanatory (Simplified Audience)

**Why it fits:**

Requests a teaching-style explanation tailored to a child (needs simple language, analogies).

**Typical output:**

Short, kid-friendly text; examples/analogies.

**3. You are a UX designer. Suggest improvements to this app layout:**

**Type:**

Instructional / Expert Advisory (Role-based)

**Reasoning:**

It starts with “*You are a UX designer*” → assigning a role. The task is to analyze and suggest improvements → consultative, expert guidance. It’s instructional because the user expects actionable recommendations on improving the app layout.

**Typical Output:**

A list of design critique points (e.g., spacing, colors, accessibility). User experience suggestions (navigation flow, readability, hierarchy). Possibly wireframe sketches or layout restructuring ideas.

## Task 2: Refinement Practice

### Prompt:

"A vibrant poster for a school gaming event, futuristic neon colors, joystick and game controller illustrations, pixel art background, energetic and fun design, bold title 'School Gaming Event 2025', space for date, time, and venue details."



**Refined Prompt:**

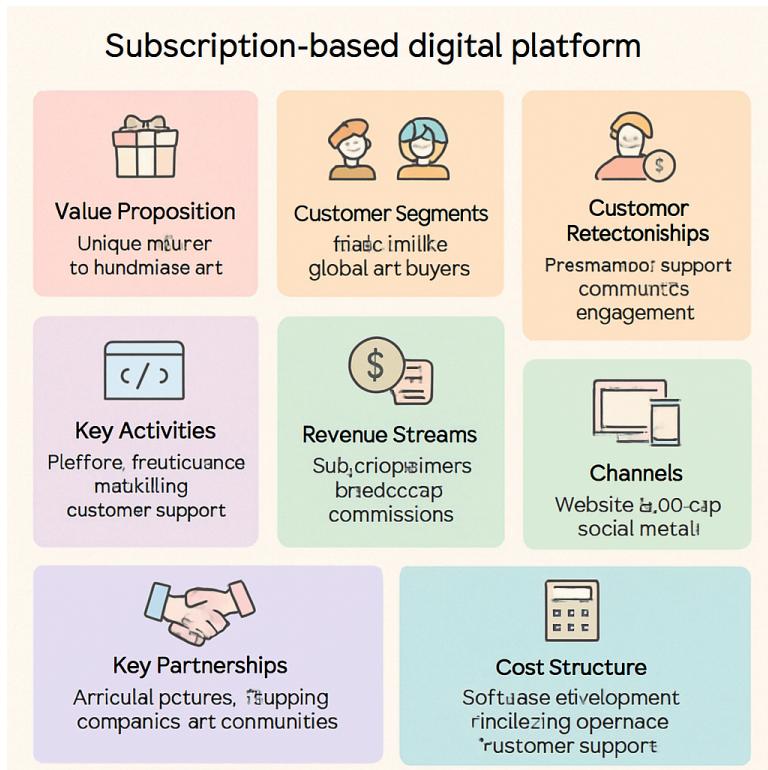
*"Refine the school gaming event poster with a clean, balanced layout. Keep the neon gaming vibe but reduce the glow for readability. Make the event title bold and central, highlight the date, venue, and time with larger clear fonts, and add playful pixel-art gaming icons (joystick, controller, coins). Include a small space for the school logo at the top."*



### Task 3: Prompt Design Exercise

#### 1. ChatGPT (text-based prompt):

"Imagine you are a business consultant helping a student entrepreneur who wants to launch a subscription-based digital platform that connects local artists with global buyers. Write a detailed step-by-step business model strategy, including value proposition, customer segments, potential revenue streams, and a 6-month go-to-market plan."



## 2. DALL·E (image-based prompt):

"Create a futuristic cityscape where traditional street food vendors are integrated into high-tech food pods. The scene should blend the warmth of human interaction with holographic menus and drone delivery in the background, using a semi-realistic art style with bright evening neon colors."



## 3. SORA (video-based prompt):

"Produce a 30-second educational video explaining the greenhouse effect to high school students. Use an engaging narrator voiceover, simple animated visuals of Earth's atmosphere, sunlight entering, heat

getting trapped, and a side-by-side comparison of a balanced vs. overloaded greenhouse gas scenario. End with a call to action: simple ways students can reduce their carbon footprint."

link:blob:<https://www.perplexity.ai/f0873156-709f-40e1-a9e8-6283be9fb134>

#### **4. Coding or Logic Prompt:**

"Write a C++ program that simulates a library management system with the following features: adding new books, issuing books to students, returning books, and displaying all available books. Use object-oriented principles with classes for Book and Student, and implement a menu-driven system for interaction."

```
#include <iostream>
```

```
#include <vector>
```

```
#include <string>
```

```
using namespace std;
```

```
// Book class
```

```
class Book {
```

```
private:
```

```
    int id;
```

```
    string title;
```

```
    string author;
```

```
    bool isIssued;
```

```
public:
```

```
    Book(int id, string title, string author) {
```

```
        this->id = id;
```

```
    this->title = title;
    this->author = author;
    this->isIssued = false;
}

int getId() const { return id; }

string getTitle() const { return title; }

string getAuthor() const { return author; }

bool getIsIssued() const { return isIssued; }

void issueBook() { isIssued = true; }

void returnBook() { isIssued = false; }

void display() const {
    cout << "ID: " << id << ", Title: " << title << ", Author: " << author
        << ", Status: " << (isIssued ? "Issued" : "Available") << endl;
}

};

// Student class

class Student {

private:

    int id;

    string name;
```

```
vector<int> issuedBooks; // store book IDs

public:

Student(int id, string name) {

    this->id = id;

    this->name = name;

}

int getId() const { return id; }

string getName() const { return name; }

void issueBook(int bookId) {

    issuedBooks.push_back(bookId);

}

bool returnBook(int bookId) {

    for (auto it = issuedBooks.begin(); it != issuedBooks.end(); ++it) {

        if (*it == bookId) {

            issuedBooks.erase(it);

            return true;

        }

    }

    return false;

}
```

```
void displayIssuedBooks() const {
    cout << "Books issued by " << name << ": ";
    if (issuedBooks.empty()) {
        cout << "None";
    } else {
        for (int id : issuedBooks)
            cout << id << " ";
    }
    cout << endl;
}
};
```

```
// Library Management System main class
```

```
class Library {
private:
    vector<Book> books;
    vector<Student> students;
    int nextBookId = 1;
    int nextStudentId = 1;
```

```
public:
```

```
void addBook() {
    string title, author;
```

```
cin.ignore();
cout << "Enter book title: ";
getline(cin, title);
cout << "Enter author name: ";
getline(cin, author);

books.push_back(Book(nextBookId++, title, author));
cout << "Book added successfully!\n";
}
```

```
void addStudent() {
    string name;
    cin.ignore();
    cout << "Enter student name: ";
    getline(cin, name);

    students.push_back(Student(nextStudentId++, name));
    cout << "Student added successfully!\n";
}
```

```
Book* findBookById(int id) {
    for (auto& book : books) {
        if (book.getId() == id)
            return &book;
```

```
    }

    return nullptr;
}
```

```
Student* findStudentById(int id) {

    for (auto& student : students) {

        if (student.getId() == id)

            return &student;

    }

    return nullptr;
}
```

```
void issueBook() {

    int bookId, studentId;

    cout << "Enter Book ID to issue: ";

    cin >> bookId;

    cout << "Enter Student ID: ";

    cin >> studentId;
```

```
Book* book = findBookById(bookId);

Student* student = findStudentById(studentId);
```

```
if (!book) {

    cout << "Book not found.\n";
```

```
    return;  
}  
  
if (!student) {  
    cout << "Student not found.\n";  
    return;  
}  
  
if (book->getIsIssued()) {  
    cout << "Book is already issued.\n";  
    return;  
}  
  
book->issueBook();  
student->issueBook(bookId);  
  
cout << "Book issued successfully
```

##### **5. Education or Training Prompt:**

"Design a micro-course module for teaching first-year college students the basics of critical thinking. The module should include: a short real-world problem scenario, three guided discussion questions, one short interactive activity (like class debate or quick case study), and a reflective takeaway exercise."

# CRITICAL THINKING MICRO-COURSE

## Highway Through a Park



### Discussion Questions

- What are the potential impacts of the highway on the park ecosystem?
- Who are the stakeholders involved and their perspectives?
- What alternatives could address the problem?



### Reflective Writing

Summarize your position and reasoning



### ● Practice Activity (Live Testing):

#### 1. Coding Prompt (for ChatGPT)

Prompt:

*"Write a Python program that calculates the factorial of a given number using recursion and provides error handling for invalid inputs."*

Generated Output:

```
def factorial(n):  
    if not isinstance(n, int) or n < 0:  
        raise ValueError("Input must be a non-negative integer.")  
  
    if n == 0 or n == 1:  
        return 1
```

```
else:
```

```
    return n * factorial(n - 1)
```

```
try:
```

```
    num = int(input("Enter a non-negative integer: "))
```

```
    print(f"Factorial of {num} is {factorial(num)}")
```

```
except ValueError as e:
```

```
    print("Error:", e)
```

Analysis:

This code uses recursion as requested, handles invalid inputs gracefully, and provides clear user interaction. It is simple and easy to understand for learners.

## 2. Video Creation Prompt (for SORA or similar AI video tool)

Prompt:

*"Create a 45-second educational video explaining the water cycle to middle school students. Include animated visuals of evaporation, condensation, precipitation, and collection, with a friendly narrator and upbeat background music."*

Generated Output:

(An animated video showing sun-driven evaporation from lakes, clouds forming condensation, rain falling as precipitation, and water collecting into bodies of water. The narrator uses simple language, emphasizing the continuous cycle, with upbeat music to keep engagement high.)

Analysis:

The video visually and audibly breaks down the water cycle into understandable parts, engaging the target age group effectively.

### **3. Education/Training Prompt (for course design)**

Prompt:

*"Design a quick workshop session to teach workplace communication skills. Include a role-play activity, three key communication tips, and a reflection question."*

Generated Output:

Workshop Outline:

- Tips:
  1. Listen actively and avoid interruptions.
  2. Use clear and concise language.
  3. Give and receive feedback constructively.
- Role-Play Activity:

Participants pair up to role-play a scenario where one must communicate a project delay to a supervisor, practicing empathy and clarity.
- Reflection Question:

"How did you feel during the role-play? What communication strategies helped you get your message across?"

Analysis:

This workshop plan balances theory and practice, promotes empathy through role-play, and encourages self-awareness with reflection, suitable for quick learning sessions.

