Title : Project Description / Project Concept Note

| SRN.No. | 202201727 |
| --- | --- |

**VISHWAKARMA UNIVERSITY, PUNE**

University Grants Commission (UGC) Approved State Private University

**Project Title: Development of a Macro Processor for Assembler Directives**

**Project Objective:** The objective of this project is to design and implement a macro processor that efficiently processes assembler directives. The macro processor will read assembler code, identify macro definitions, expand them, and output the resulting code. The system will be designed to handle standard assembler directives, support nested macros, and provide an easy-to-use interface.

**Project Outcome:** The outcome of the project will be a fully functional macro processor capable of reading an assembly program containing macro definitions and directives, expanding the macros, and generating an output file with the expanded assembly code. The processor will enhance the efficiency of writing assembly language programs by automating repetitive tasks through macros.

|  | **Project Description** | **Marks** | **CO** | **BTL** |
| --- | --- | --- | --- | --- |
| **1** | **Definition** | **15** |  |  |
|  | A macro processor is a tool that simplifies assembly language programming by allowing the definition of macros. Macros are sequences of instructions that can be reused multiple times, saving time and effort. The macro processor reads assembler directives, expands the macros, and generates the final assembly code. |
| **2** | **Methodology** |
|  | **Requirements Gathering**:   * Identify the key functionalities required for the macro processor. * **Design**: Create a basic architecture for the macro processor, including components for macro definition, invocation, and processing assembler directives. * **Implementation**: Develop the macro processor using a suitable programming language with a simple graphical user interface (GUI) to facilitate user interaction. * **Testing**: Test the macro processor with various assembler directives to ensure it works correctly and efficiently. * **Documentation**: Document the design, functionality, and usage of the macro processor. |
| **3** | **Implementations** |
|  | The implementation of the macro processor involves the creation of a graphical user interface (GUI) that allows users to interact with the processor easily. The main components include a text editor for writing assembly code, options for defining macros, and a compilation process that processes assembler directives. The macro processor parses the input code, identifies macros, and replaces them with their expanded forms during compilation. Features such as error handling and syntax highlighting are integrated to enhance the user experience. Additionally, the processor maintains a symbol table to keep track of defined macros, enabling efficient macro expansion. The project leverages object-oriented programming principles to ensure modularity and reusability, allowing future enhancements and adaptations to different assembler languages. By the end of the project, users will have access to a robust tool that simplifies assembly programming through effective macro management. |
| **4** | **Output & Evidence** |
|  |  |

**Course Outcomes**

| **CO No.** | **Statement** |
| --- | --- |
| 1 | Discriminate among different System software and their functionalities. |
| 2 | Design language translators like Macro processor and Assembler. |
| 3 | Develop approaches and methods for implementing compiler, linker and loader. |
| 4 | Use LEX tool for lexical analysis. |
| 5 | Interpret the techniques of implementing utility software. |

**Bloom’s Taxonomy Level (BTL)**

| **BTL No.** | **BTL** | **Statement** |
| --- | --- | --- |
| 1 | Remember | Recall facts and basic concepts |
| 2 | Understand | Explain ideas or concepts |
| 3 | Apply | Use the information in new situations |
| 4 | Analyze | Draw connections among ideas |
| 5 | Evaluate | Justify a stand or decision |
| 6 | Create | Produce new or original work |