**INTRODUCTION**

* 1. **Existing System**
* There are a number of existing systems for this application but are much advanced having many features that will be unused by beginner programmers and difficult for beginner to learn.
* So we are creating a small application that will be useful for beginner programmers and help them to learn things faster.
  1. **Problem Definition- Need of Computerization**
* Programmers have to write the whole skeleton of programs and then edit it using their logic which is time consuming, if we have an application that can create the skeleton of the program by just taking the class name, number of variables and their types, number of function and their arguments, inheritance, etc.
* It will be easier for the programmer to edit the program as per their logic and easily track if any other function or class is remaining for applying the logic.
* Usually programmers start their programs building class and if any function or module is remaining to declare or define the program results in logical error or compilation error making it difficult to debug.
* So the application will give a modular approach by declaring all the class and its aspects to just be modified by the programmer, hence making a programmers work easier and efficient.
* New upgraded modules can be made available or might be already available.

**PROPOSED SYSTEM**

**2.1 Proposed System**

* A Stand Alone Application that will generate a Skeleton of the Program on Basis of Input.
* The Application will be Java Based Application and will generate Java Code Skeletons.
* It will give the User a Skeleton of above Input which can be edited as per the User’s Logic.

**2.2 Objectives of System**

* The Application will take Following as Input:

1. Number of Classes in the Code.
2. Class name.
3. Inheritance (if any).
4. Variable Types and Number of Variables for the Types.
5. Constructor and Destructor (if any).
6. Number of Methods and their Arguments its Type and Return Type of the methods.
7. If the Class contains main() function.

* The Application will have Following Output:

1. Code for previously Created Programs.
2. Code for input Program.

**2.3 Feasibility Study**

**2.3.1 Technical Feasibility**

This Feasibility Study has important step in every software development process because it makes analysis of different aspects that we require in terms of hardware and software to run the application without any error and with efficiency.

1. Software Requirements:
2. Our System can use any operating system that supports Java and have jre, as jre are available free we do not need to pay any amount.
3. Our system creates files to store the output data.
4. Our system uses Java which creates a user friendly GUI to make application flexible and easy to use, as Java is freeware and we are not using any paid packages we do not need to pay any amount.
5. NetBeans IDE t code the desired application.
6. Hardware Requirements:

Our system will require the following minimum hardware requirements to run the application efficiently:

1. Processor should be at least Pentium 4.
2. At least 512 MB of RAM to run application efficiently and lag free.
3. At least 500 MB of free space to install and run.

As our hardware and software requirements are easily available. Therefore, our system is technically feasible.

**2.3.2 Operational Feasibility**

This Feasibility study is important step as it focuses on the user interface. It covers whether there is any kind of training required to use the application or does the user need to know the complex design of the application.

Our system uses Java to give the application its dynamic interface.

Therefore, there is no need of training as the application will be self-navigating. It has a user interface normal applications have, follows the basic standards of structure design hence, it is easy to use and has friendly user interface.

Hence no training will be required to use the system, our system will be operational feasible.

**2.3.3 Financial Feasibility**

This Feasibility study is important as it focuses on the capital to be invested in the development of the application. It estimates whether the capital investment is more or less.

Our system will require Java, JRE and OS to run the application.

The above software requirements are open source and are easily downloaded free of cost.

The hardware required is easily available in market and is standard specification.

Hence, as our system will use open source, no paid packages, and standard hardware available in the market in development of the application. Our system is economically feasible.

**USER MANUAL**

**4.1User Manual**

The application is coded in Java language. It is having a user friendly interface. So run it and you are ready to go.

**4.2Menu Explanation**

Window 1:

There will three options:

1. New:

To create a new class by providing all inputs required.

1. View:

To view the existing files created during previous operations.

1. Exit:

Exit application.

Window 2:

1. Number of Classes in the Code.
2. Class name.
3. Inheritance (if any).
4. Variable Types and Number of Variables for the Types.
5. Constructor and Destructor (if any).
6. Number of Methods and their Arguments its Type and Return Type of the methods.
7. If the Class contains main() function.

Click on Generate and file will be created having skeleton of code using given information.

**CONCLUSION**

**5.1 Limitations & Drawbacks**

Like any other research, this study also has some limitations.

* The system currently only work with classes, Interface is not included implicitly.
* If user wants Interface included, user will have to manually modify the file.
* Once user gives all inputs and file is created it cannot be again modified using application.
* There is no probation to create packages.
* No probation to import inbuilt packages.

**5.2 Future Enhancements**

* The goal of this project is to produce an interactive and user-friendly application for developers to reduce their effort in coding.
* The application will include probation to modify the file after creation according to user requirement using application.
* It will also include features like import built-in packages, create packages and Interface.

**5.3 Conclusion**

* The Code Skeleton generator (CSG) system was made to meet interactive, visual menu that provides the user to code in completely new manner.
* It reduces efforts of developer to code and gives a skeleton of whole code user wants and only need to put logic in that code.
* User gets interactive environment to input the data.
* This application will be very help to create large code files.
* Beginners will get beneficial due to its simple interface.

**5.4 References & Bibliography**

#### BOOKS:

* Grady Booch, James Rambaugh, The Unified Modeling Language User/Reference Guide, Pearson Education INC
* Complete reference Java by Herbert Schildt(5th edition)
* Software Engineering: A Practitioner’s Approach (Seventh Edition) by Roger S. Pressman, McGraw Hill International Edition.

**WEBSITE:**

* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* [www.oracle.com](http://www.oracle.com)