



RAKs. M. Plus an Efficient Computer Programming Language for Future Advancement Scenario

Rakesh Kumar^{1*}

¹Department of Computer Science and Engineering, Cambridge Institute of Technology, Tatisilwai, Ranchi-835103, Jharkhand, India.

Author's contribution

The sole author designed, analysed, interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/JAMCS/2018/16484

Editor(s):

- (1) Dr. Dariusz Jacek Jakóbczak, Assistant Professor, Chair of Computer Science and Management in this department, Technical University of Koszalin, Poland.
(2) Dr. Tian-Xiao He, Professor, Department of Mathematics and Computer Science, Illinois Wesleyan University, USA.

Reviewers:

- (1) Trong Wu, Southern Illinois University Edwardsville, USA.
(2) Suraya Masrom, Universiti Teknologi MARA, Malaysia.
(3) Amal Abd El-Raouf, Southern CT State University, USA.

Complete Peer review History: <http://www.sciedommain.org/review-history/26810>

Received: 03 February 2015

Accepted: 20 April 2015

Published: 24 October 2018

Original Research Article

Abstract

Since there are many procedural high level languages exists such as C, C++ etc. though we found many issues or drawbacks [1] that make these languages inefficient and slow in term of processing the real scenario programming. In this paper we have discussed why RAKS M. PLUS is better, efficient and more secure than other programming languages by explaining the working mechanism of this programming language. With the help of this paper we put your effort to focus on those issues that make the processing slow of any programming and how by taking a step that make the programming processing fast for the future deployment and advancement point of view. Also we try to create a new programming language RAKS. M. PLUS which deals with security problems to make it more efficient. Comparative study and analysis using C, C++ language platform [1]. Department of Computer Science and Electronics & Communication Engineering, Since December 2012 to December 2014 in Simulation Lab, Cambridge Institute of Technology, Ranchi, Jharkhand, India. Since this language provides an environment for designing the kernel RAK M. PLUS which will provide fast compilation or processing of real scenario programming as well as wallet to store some user beneficial details. Reverse engineering followed to recreate the program and also to weak the existing deficiencies. Important steps against decomposition process used in the program to understand the code functionality, correcting code and also to secure the performance smartly. Apart from that improved SQL injection method used in web application for

*Corresponding author: E-mail: rsharma.ranchi2009@gmail.com;

security vulnerability. Inbuilt function has been used instead of header file and this will help to decrease the execution time up to 40-50 percentages as compared to Turbo / Borland C++. The result will be purely depend upon user's application and improves the performance level of execution. This language proves to be efficient in terms of time consumption, fast, secure and user friendly from student to scholar level.

Keywords: RAKs. M. PLUS; C; C++; security issue.

Note: User may use almost all the keywords of C++ with a slight change in the name. `_rmp` is added to the name of keywords. For example, if the user wants to use the keyword `getch()` then it should be rewritten as `getch_rmp()`; The main purpose of doing so, is to make the language acceptable and easy to learn by the users of RAKs. M. PLUS.

1 Introduction

Definition: RAKs. M. PLUS is a partially object oriented computer programming language which is designed to overcome the drawbacks of Turbo / Borland C++. RAKs. M. PLUS is nearer to C++. There is a slight difference in the keywords of C++. A suffix “`_rmp`” is added to all the keywords or pre-defined (i.e. inbuilt functions). For example if the user wants to use “`cout`” in RAKs. M. PLUS, the user should write “`cout_rmp`”. The major changes made in the RAKs. M. PLUS computer programming language is in the internal mechanism as well as in the security area. *It is not a simulation software.* Only the authorized user this computer programming language. All the powers are given to the admin. The admin has the power to accept or reject the request for master password to retrieve the stored files. No user can access the files of another users as the files will be in encrypted form. RAKs. M. PLUS supports all the four main pillars of Object Oriented Programming i.e. Class and Object, Encapsulation, Polymorphism and Inheritance and many other features also. It will provide an environment for designing the kernel. RAKs. M. PLUS language is expected to be faster than C++ and provide very high security environment. It cannot be hacked at user level. The RAKs. M. PLUS can be used for problem solving, for designing kernel as well as a wallet to store important information such as account no, pin code, email-id, passwords etc.

2 Pillars of Object Oriented Programming

There are four main pillars of Object Oriented Programming which are as follows:

2.1 Abstraction [2]

The concept of abstraction relates to the idea of hiding data that is not needed for presentation. The main idea behind data abstraction is to give a clear separation between properties of data type and the associated implementation details. This separation is achieved in order that the properties of the abstract data type are visible to the user interface and the implementation details are hidden. Thus, abstraction forms the basic platform for the creation of user-defined data types called objects. Data abstraction is the process of refining data to its essential form. An Abstract Data Type is defined as a data type that is defined in terms of the operations that it supports and not in terms of its structure or implementation.

2.2 Encapsulation [2]

Encapsulation is the process of combining data and functions into a single unit called class. Using the method of encapsulation, the programmer cannot directly access the data. Data is only accessible through the functions existing inside the class. Data encapsulation led to the important concept of data hiding. Data hiding is the implementation details of a class that are hidden from the user. The concept of restricted access led programmers to write specialized functions or methods for performing the operations on hidden

members of the class. Attention must be paid to ensure that the class is designed properly. Neither too much access nor too much control must be placed on the operations in order to make the class user friendly. Hiding the implementation details and providing restrictive access leads to the concept of abstract data type. Encapsulation leads to the concept of data hiding, but the concept of encapsulation must not be restricted to information hiding. Encapsulation clearly represents the ability to bundle related data and functionality within a single, autonomous entity called a class.

2.3 Polymorphism [2]

Polymorphism is the ability to use an operator or method in different ways. Polymorphism gives different meanings or functions to the operators or methods. Poly, referring too many, signifies the many uses of these operators and methods. A single method usage or an operator functioning in many ways can be called polymorphism. Polymorphism refers to codes, operations or objects that behave differently in different contexts.

2.4 Inheritance [2]

Inheritance is the process by which new classes called derived classes are created from existing classes called base classes. The derived classes have all the features of the base class and the programmer can choose to add new features specific to the newly created derived class. This concept of Inheritance leads to the concept of polymorphism. Inheritance helps the code to be reused in many situations. The base class is defined and once it is compiled, it need not be reworked. Using the concept of inheritance, the programmer can create as many derived classes from the base class as needed while adding specific features to each derived class as needed. The above concept of reusability achieved by inheritance saves the programmer time and effort.

3 Terminology Followed

3.1 Reverse Engineering [3]

Reverse engineering is a technique used to analyze software in order to identify and understand the parts it is composed of. The usual reasons for reverse engineering a piece of software are to recreate the program, to build something similar to it, to exploit its weakness or strengthen its defenses.

Because closed, proprietary software never comes with documentation that reveals the source code used to create it, people use reverse engineering whenever they want to understand the software's inner workings.

Some hacker use reverse engineering to find weak points of programs which they can exploit. Other hackers use reverse engineering to locate weak points with the intension of strengthening the defenses there.

3.2 Decompilation [4]

Decompilation is the process of transforming a binary executable i.e. a compiled program into a higher level symbolic language that is easier for humans to understand. Usually this means turning a executable program into source code in a language like C. Most systems for decompiling cannot directly convert program into 100% source code. Instead they usually provide an "almost there" kind of intermediate representation. Many reverse compilers are actually de-assembler that provide a dump of the machine code that makes a program work.

Probably the best decompiler that available to the public is called IDA-pro. IDA starts with a disassembly of program code and then analyzes program flow, variables, and function calls. IDA is hard to use and requires

advanced knowledge of program behavior, but its technical level reflects the true nature of reverse engineering. IDA supplies a complete API for manipulating the program database so that users can perform custom analysis.

Decompilation is applied for multiple reasons, including:

- Understanding code functionality
- Correcting errors
- Enhancing computer security
- Removing restrictions, such as passwords, copy protection, and time limit.
- Recovering lost source code for achieving purposes

3.3 SQL injection [5]

SQL Injection is a type of web application security vulnerability in which an attacker is able to submit a database SQL command which is executed by a web application utilizes user supplied data without proper validation or encoding as a part of a command or query. SQL Injection allows an attacker to create, read, update, alter, or delete data stored in the back end database. In its most common form, a SQL Injection attack gives access to sensitive information such as social security numbers, credit card number or other financial data.

3.4 One-pass compiler [6]

A one-pass or single pass compiler is a type of compiler that passes through the parts of each compilation unit only once, immediately translating each part into its final machine code. This is in contrast to a multi-pass compiler which converts the program into one or more intermediate representations in steps between source code and machine code, and which reprocesses the entire compilation unit in each sequential pass.

3.5 Context-free grammar [7]

In formal language theory, a *Context-Free Grammar (C.F.G)* is a formal grammar in which every production rules is of the form

$$V \rightarrow w$$

Where V is a single non terminal symbol, and w is a string of terminals (w can be empty). A formal grammar is considered "Context-Free" when its production rules can be applied regardless of the context of non-terminal. No matter which symbols surround it, the single non terminal on the left hand side can always be replaced by the right hand side [8].

3.6 Acceptability or compatibility

The acceptability or compatibility of this language is that it supports all the exiting operating system and works on existing 32 bit version such as Window 95/98/NT/2000/XP/Vista. It can be compiled with Visual C++ 4.x/5.x/6.x or with the Borland C++ 5.x compiler. You may also need to determine which compiler has been used to compile the various scripting languages that you will be using.RAK M.PLUS is implemented in C and C++ and is distributed in source form. You will need a working C++ compiler (e.g. g++) to build this language and at least one of the supported scripting languages to use it (or else it isn't going to be very useful). RAKS M.PLUS does not depend upon any of the supported scripting languages for its own compilation. Finally, although RAK M.PLUS is partly written in C++, a C++ compiler is not required to use.

4 Advantages and Limitations of Object Oriented Programming [8]

4.1 Advantages

1. Improved software-development productivity
2. Improved software maintainability
3. Faster development
4. Lower cost of development
5. Higher-quality software

4.2 Limitations

1. Steep learning curve
2. Larger program size

5 Importance for Future Scenario

- a. It will provide very high security which cannot be hacked at user level either by DECOMPILING, REVERSE ENGINEERING or by SQL INJECTION.
- b. We do not have to put any 'header file' in RAKS. M. PLUS programs.
- c. The execution time of RAKS. M. PLUS program will 40-50 % less as compared to Turbo / Borland C++.
- d. It will have a number of inbuilt functions which can be used by user to perform the task comparatively easy.
- e. Size of the RAKS. M. PLUS program will be short.
- f. Only authorized user can access and no one can see the contents of the files of other user not even admin. Admin can only reset the password.

6 How to Use This Efficient Language

1. Firstly users have to create an account. Admin will verify the entered details. If found correct, he/she will give a security code of 4-5 digit which is when entered correctly, account will be created.
2. Users then enter the login details to use RAKS. M. PLUS.
3. If anyone enter wrong password three times continuously, his/her account will be deleted.
4. To retrieve the account user has to enter security code as well as the answer of security question.
5. After login, start coding. To save the program, enter your password.
6. To view the list of files, user will be asked to enter admin password.

*Note: The saved file will be in encoded form. To decode it, the user will have to enter the same password which was set at the time, some user created it.

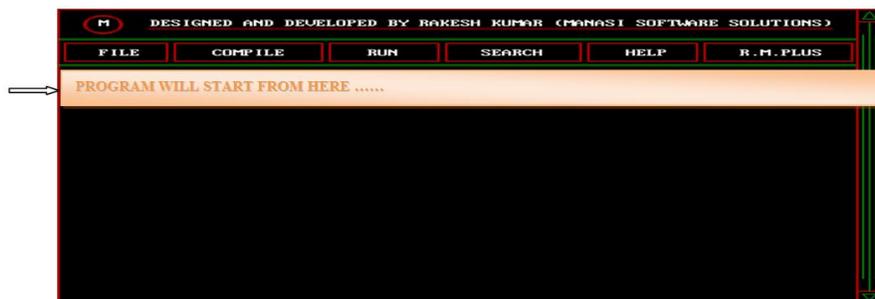


Fig. 1. Platform designed to run language

If a user is unable to retrieve account, admin may recover and reset password by entering a master password of length 50.

7 Advantages and Limitation of Proposed Language

7.1 Advantages

Mention below is some of the advantages of proposed language

1. There will be no need to put the header files in the beginning of RAKS M. PLUS Program.
2. Its compile time will be less.
3. There will be a lot of inbuilt functions in RAKS M. PLUS, with the help of which user may easily perform the task.
4. It will provide very high security so, no other user can access the file of other user as the saved file will be in encoded form.
5. No ambiguity problems will occur at the time of execution (i.e. run time).
6. Exception Handling or error handling will be easily possible.
7. Array Bound Check - size of the array will be checked, so that it will not read the extra elements.

7.2 Limitations

Mention below is some of the limitation regarding proposed language

1. We have to declare all the variable just after main()
2. Mouse will not work in RAKS M. PLUS.

*Note: These two limitations can be easily resolved in future.

8 Structure of Proposed Language

```
main ()
{
    Declaration of variables;
    Expression and Operations (Body)
}
```

Consider the following block of code:

Example 1.0: RAKS M. PLUS program to display a name on the screen.

```
main ()
{
    cout_rmp<<"RAKESH KUMAR SHARMA";
    getch_rmp ();
}
```

Output of the above example:

RAKESH KUMAR SHARMA

RAKs. M. PLUS program starts with main() function. There is no need to mention any header file. Then the user has to declare the variables which are to be used in the program. Here there is no variables required in

the above program. *cout_rmp* is the keyword which is used for displaying. And finally *getch_rmp()* is used to hold the output screen.

The internal mechanism that we follow is:

We will create a Database with the help of interfacing C++ with My SQL.

1. There will be 26 tables. Each table will contain set of keywords (pre-defined functions) starting with a particular alphabet. (separate tables for each alphabet)
2. There will be one table in the Database which will contain all the pre-defined constants.
3. There will be a table which will maintain the user details and their login credentials. This table cannot be updated manually or deleted.
4. There will be the master table which cannot be updated or deleted until admin enters the Master Password.

*Note: During implementation of RAKS M. PLUS, we will define all the functions in the same source file.

Following are used for the implementation of RAKS. M. PLUS:-

1. SQL Constraints - for security reasons.
2. Single Pass Compiler – Own Single Pass Compiler is used as it may reduce the compile time of RAKS. M. PLUS programs and due to its different advantages over the multi pass compiler.
3. Context Free Grammar (C.F.G) - we will use context free grammar during multi pass compiler construction.

Consider the above program:

First of all, compiler will check the whole program for errors. If found no error, it will display “Compiled Successfully!!!”. Else it will display a message “Errors Found Press any key to continue.” After then display the errors.

*Note: Here there is no error in the above program. The keyword “*cout_rmp*” when encountered, *cout_rmp* () function will be called. Similarly when *getch_rmp* () is found, *getch_rmp* () function will be called. After execution, it will display “RAKESH KUMAR SHARMA”. How to use this language is shown in term of flow chart as (Fig. 2).

9 Security Issue

For high security, we will maintain following files:

1. A file which will keep the login credentials, the “Master Password” and a copy of that file.
2. A file which will maintain the user details. For instance, which user logged on at which time and for how much time? If a user logged on with login id as RS365 at 12:30 P.M. for 30 minutes on 5th Sept 2014. It will store the record as follows:

Sl. no.	Date	Login ID	Time	Used time (in minutes)
1.	05/09/2014	RS365	12:30 P.M.	30

Note: To keep the login credentials safe, we will have to do register programming, so that it cannot be hacked at the user level even by back tracking. At the time, when users enter the login credentials, both login id and password will be displayed as “”. If a user enters wrong password for three consecutive times his/her account will be deleted from the main file (but not from the backup file). To recover account, the user has to

answer security question. If user failed to recover the deleted account then to retrieve the file which is saved by him/her, user has to ask the admin for the “Master Password”.

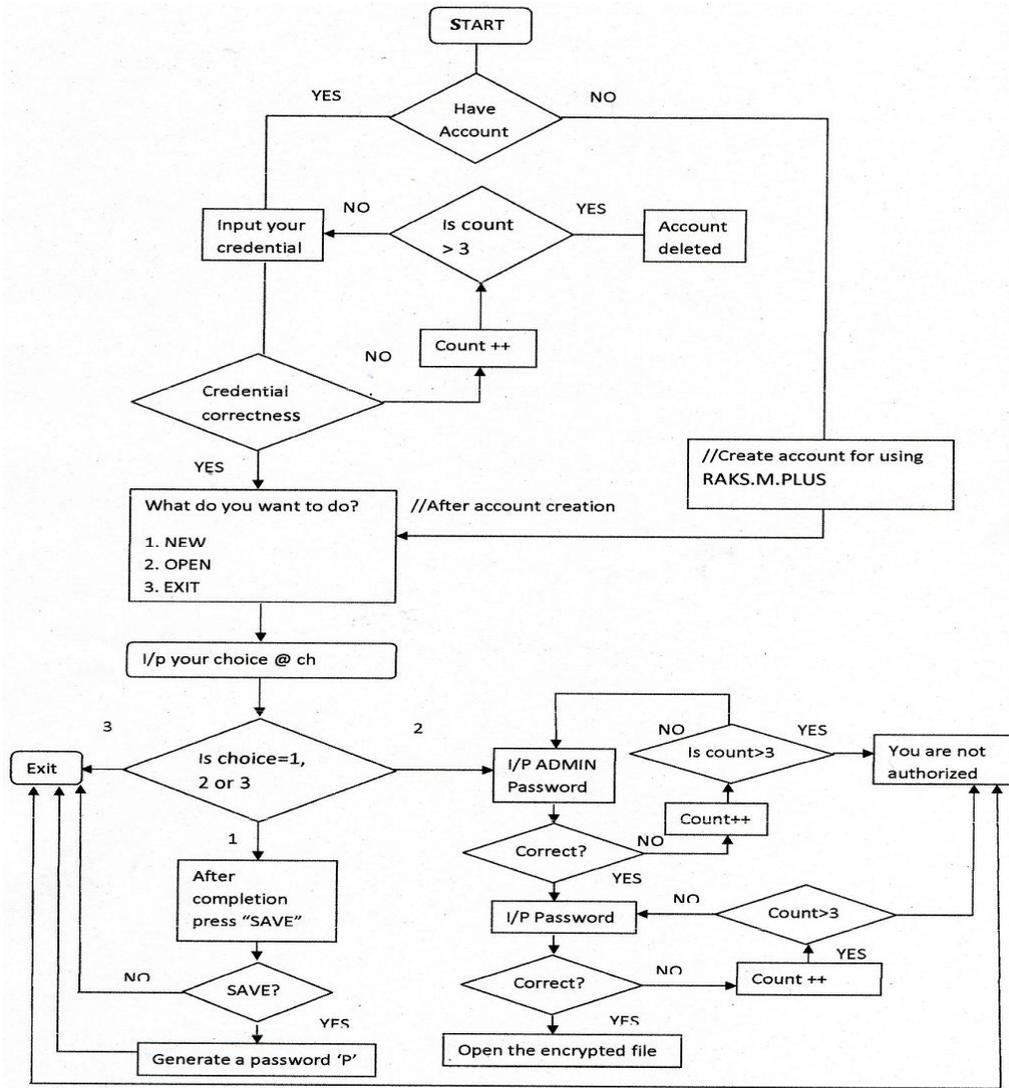


Fig. 2. How to use this language

10 How This Language Faster than C++

RAKS M. PLUS is faster than C++ because of the following:

1. We are in lining Assembly Code to the definition of functions during implementation.
2. We are using the hashing property to DETECT the keywords during implementation.

Due to the above two reasons the compile time and execution time of RAKS. M. PLUS is less than that of C++.

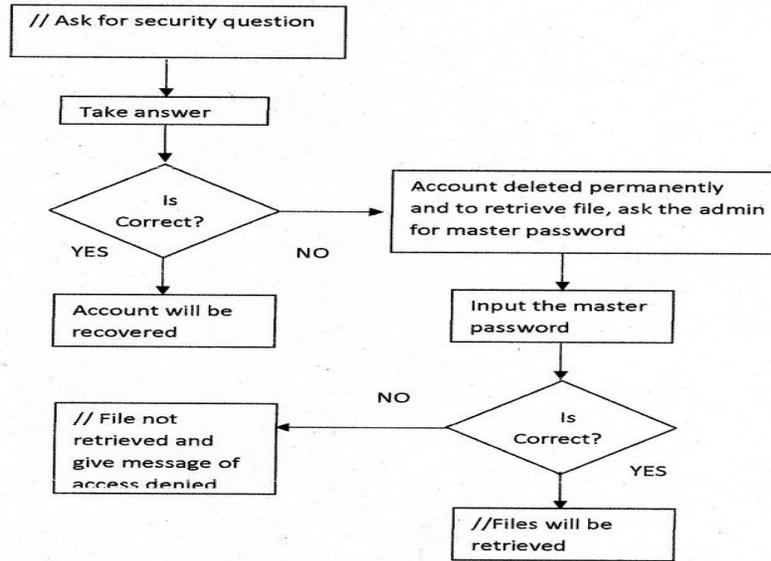


Fig. 3. How to recover account

11 Conclusion and Future Scope

If we implement this language, it may proved to be an efficient programming language which will provide a very high security and will be user friendly with comparatively less execution and compile time. It may bring a drastic change in the world of Computer Programming Language.

We can extend the features of RAKS M. PLUS by including the inbuilt functions for animation and introducing features for designing a web page. More over we may implement the Database Concepts which will keep a collection of data or information and allows performing several operations on that.

Acknowledgement

I am thankful to Prof. Shrikant Upadhyay faculty of Electronics & Communication Engineering Department, Cambridge Institute of Technology, Ranchi for his technical support and guidance.

Competing Interests

Author has declared that no competing interests exist.

References

- [1] Available:http://locklessinc.com/articles/10_problems_with_cpp/
- [2] Available:<http://www.kbcafe.com/articles/OOP.Concepts.pdf>
- [3] Available:<http://www.techopedia.com/definition/3868/reverse-engineering>
- [4] Available:<http://www.debugmode.com/dcompile/>

- [5] Available:<http://www.veracode.com/security/sql-injection>
- [6] Available:https://en.wikipedia.org/wiki/One-pass_compiler
- [7] Available:http://en.wikipedia.org/wiki/Context-free_grammar
- [8] Available:www.saylor.org/site/wp-content/uploads/2013/02/CS101-2.1.2-AdvantagesDisadvantagesOfOOP-FINAL.pdf

© 2018 Kumar; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)

<http://www.sciencedomain.org/review-history/26810>