RESEARCH METHODOLOGY AND RESEARCH DESIGN

1. What is Java?
2. Discuss about Editor of java IDE like Eclipse, Netbeans etc. of JAVA.
3. Versions of Java.
4. Structure of Java Program.
5. Discuss about Memory Management and Compilation Program.
6. Discuss about API.
7. Discuss of Third party tools.
8. Discuss of Various Database Connectivity with Java Language.
10. Discuss about application and Project or software and implements in Java.
11. Future of JAVA.

HYPOTHESIS

Research work mentioned all the aspect relative to Java Programming Language. This thesis is an attempt to investigate how students develop project and applications in Java and understandings of all concepts in Programming disciplines. In order to have a reasonable focus, its scope is limited to learning computer programming, and centrally, coming to understand the concepts of object-oriented programming (OOP) in Java. It is restricted to novices – students with little or no previous experience of programming.

It is not about the reasoning processes of professional programmers. I hereby promise to do considerable amount of work, from an essentially ergonomic point of view, with the motivation of making software development as effective as possible. This thesis is about learning programming rather than professional practice.

Neither is it about problem-solving. It is obvious that programming, both by novices and experts, contains some element of problem-solving, whether it is radically innovative or just the recognition of a context and the application of routine techniques. But the focus here is on students coming to understand the foundational concepts of OOP, rather than original uses of them to solve problems.
The approach tries to be as fundamental as possible. This derives from the assumption that people understand Java the same way that they understand anything else. Human cognition is assumed to be a consequence of evolutionary pressure which has resulted in neural and social structures which enable us to „make sense of a remarkable range of ideas. It is unreasonable to suppose that students have special cognitive apparatus devoted to Java and OOP. Consequently it is necessary to look at how humans understand concepts in general. Java is simply the context.

Why Java?
While some investigation in this thesis looks at general notions of computers and the execution of short pieces of pseudo-code, the focus will be on the Java language and associated OOP concepts. There are several reasons for this. Java is currently a very widely used language, for desktop applications, client-server systems (using „Java Enterprise Edition) and mobile phone applications on Android.

In the face of the best efforts of teachers in our discipline, many students are still challenged by programming. A 2001 ITiCSE working group (the “McCracken group”) assessed the programming ability of a large population of students from several universities, in the United States and other countries [McCracken, 2001]. However numerous studies have shown that many students encounter problems when learning it - such as the McCracken Group Report (2001), The ACM Java Task Force (Roberts et al 2006), Hu (2006), Fleury (2000), Holland, Griffiths and Woodman (1997), Garner, Haden and Robins (2005), and Thomasson, Ratcliffe and Thomas (2006).

One aspect of this is that Java was designed so that professional programmers accustomed to languages such as C would be able to develop proficiency in it quickly. The syntax of variable declaration, assignment statements, loops and conditionals in Java is similar or identical to that of C, and there is an assumption that the person learning Java is already familiar with these concepts, which of course is not true for novices.

A second aspect of Java is that it involves a set of inter-dependent ideas, which cannot be isolated. As a fairly „pure OOP language, even the simplest executable program makes reference to a host of unfamiliar concepts, represented by the keywords public, class, static, void and. Beyond the first program, OOP entails a number of ideas which are
distinct but closely linked, such as class and object. This presents a difficulty in designing a learning sequence, which one would ideally like to lead the student to encounter new ideas one at a time – impossible in Java because the concepts are dependent upon each other.

A third aspect is that there is a convention that classes and methods are named so as to make them self-descriptive (contrasted with languages like Perl where brevity is more important). There are clear advantages in this, but for a student there is the problem that they easily acquire a „bleached idea of what something is, and this can mask a precise understanding. For example a student may well think they „understand what Math.PI is, but not be aware that it is a final static member of the class Math. This also makes it difficult to investigate students understanding, since they can often make shallow but correct guesses as to what a program fragment means.

Fourthly, the notion of symbol has abnormal significance in Java. It is argued here that most of the content of all programming languages, and discourse about computers, is derived from metaphors and other conceptual blends. These metaphors are mostly unobserved – such as computer „memory or program „instruction. But in the case of OOP there is a further layer of more or less conscious metaphor, such as class, object, method and member. These correspond to literary metaphors, like „All the world’s a stage, where the author is aware that they are inventing a metaphorical way of looking at something. Consequently the student of Java has to deal with two layers of metaphor, even though typically neither layer is pointed out in an instructional course.

These features mean Java is an effective vehicle for exploring this thesis, and provide affordances and constraints in the research. Pragmatically Java is used because it is what most novice students are taught.

**Concepts**

What are the concepts used in OOP? Armstrong (2006) examined 239 sources published between 1966 and 2005 to identify the most commonly used set of OOP ideas. The 'top ten' were in order:

1. Inheritance
2. Object
3. Class  
4. Encapsulation  
5. Method  
6. Message Passing  
7. Polymorphism  
8. Abstraction  
9. Instantiation  
10. Attribute  

WORK PLAN AND METHODOLOGY

In this dissertation our main aim is to study and implement and make own structure or an Algorithm to improve the Security and how student develop Applications or projects using java and how many library provided for security and packages in java, how to make project or web application more robust, fast, and reliable in java and discuss various tools. Complete dissertation will be covered in a Five Chapters. The entire Chapter further divided into number of modules. Abstract of Five chapters is as follows:

Chapter 1: Introduction of JAVA Language.

In this chapter we will discuss about the various module such as:

✓ What is a Java programming Language?
✓ Overview of J2SE [Core JAVA]
✓ Overview of J2EE [Advanced JAVA]
✓ Overview of various technique available

Chapter 2: Literature Survey

In this chapter we will make survey of the Java application One tier, Two tier etc. Discuss of java application like Console, Desktop, Web, Enterprise and Mobile Application. I will be discuss the Various tools are provided by java. I will be discuss the practical on java security that which library will be providing for secure the Programs and Application.
Chapter 3: Problem Definition

In a problem definition we will discuss about the problems. In problem definition discusses about the need of research done in java design secure application and Security issues. Even though extended amount of work is already done in this area we will further find various loop holes of different technologies used in the research of java application and Security.

Chapter 4: Implementation of Techniques for Make Java Application or other.

In this chapter we will discuss about the implementation of the different technique for developing java application, Security Issues and Security tools, which will be helpful to solve the problem. While making various application and others. I have discussed about various software and IDE which will help in making application or project or software.

Chapter 5: Discussion & Result

In this chapter we will describe the complete result of the discussion. Moreover, find which technique (Software, Third Party tools etc.) are used to developing secure, robust application or software in java.