



TRAINING REPORT

OF

CO-OP TRAINING, UNDERTAKEN

AT

“WIPRO LIMITED”

IN

Project Engineer

ON

“SAP ABAP AND PROJECT READINESS PROGRAM”

SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of:
Name: Amit Monga
Designation: Asst. Professor
Department: ECE

Submitted By:
Name: Rajneesh Gulati
University Id: 1511981522

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my project supervisor, Mr. Rahul Srivastava, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.

I would also like to thank the entire Wipro Limited for being so supportive in shaping this report.

Thanking You

Rajneesh

Certificate



11th February 2019

Mr. Rajneesh Gulati
#5041 / A Sector 38/West Chandigarh, 160014

Dear Rajneesh,

Sub: Letter of Engagement as Intern

We are pleased to inform you that you have been selected for undergoing Internship in our organization Wipro Limited (Wipro) as Intern under the following terms and conditions:

1. Nature of Engagement

You will be engaged as an **Intern at Wipro.**

2. Duration of training

The duration of internship will be from **11th February 2019 to 11th August 2019**. During this period, Wipro shall evaluate your performance. Unless Wipro extends the period of internship, in writing, solely at its discretion, your internship shall automatically stand terminated at the expiry of the internship period.

3. Verification Report

Your engagement with Wipro will be subject to receipt of satisfactory report with regard to verification of the particulars furnished by you in your application and information given at the time of Interview. If any declaration or information furnished to Wipro proves to be false or if you have wilfully suppressed any material information, in such case, you will be liable to removal from training without any notice.

4. Obligations and Responsibilities

- a. During your internship period, Wipro expects you to undergo training in any department / section in which you are placed with high standard of initiative and efficiency. You shall devote yourself exclusively for undergoing training. You shall not take up any other work for remuneration (part-time or otherwise) or work on advisory capacity or be interested directly or indirectly in any other trade or business (except as share-holder or debenture holder) during the training period without obtaining permission in writing from the appointing authority at Wipro. You will be governed by the service rules / standing orders, policies and regulations as may be promulgated by Wipro from time to time in relation to conduct, discipline and other matters. You will not seek membership of any local or public bodies without first attaining specific permission from the appointing authority at Wipro. You are expected to comply with the policies of Wipro including the Code of Business Conduct and other policies of Wipro as they form an integral part of the terms of your training with Wipro.

Registered Office:

Wipro Limited
Doddakannelli
Sarjapur Road
Bengaluru 560 035
India

T : +91 (80) 2844 0011
F : +91 (80) 2844 0054
E : info@wipro.com
W : wipro.com
C : L32102KA1945PLC020800

Sensitivity: Internal & Restricted

Abstract

This Internship was aimed at providing basic theoretical knowledge about working on SAP platform which is both coding as well as other technical aspects of it. In SAP, my role was of technical consultant which included writing code in ABAP to generate business reports. Also, with duration of time I got a chance to work on C++ using Linux server under the 'Project Readiness Program'.

In the 'Project Readiness Program' we were made familiar with C++ concepts and basic Linux commands which enhanced our knowledge on C++.

Table of Contents

Sr No	Title	Page No
1	Introduction	1
2	Acknowledgement	2
3	Certificate	3
4	Abstract	4
5	Table of Contents	5
6	Purpose	6
7	Requirement Hardware	7
8	What is Linux?	8
9	About C++	12
10	Logic Building on PBLApp	15
11	Code C++	18
12	What is SAP?	21
13	ABAP	29
14	Code ABAP	32
15	References	36

PURPOSE:

The aim of the internship was to provide basic technical knowledge about the SAP platform as to how to write codes in ABAP for generating business reports working on tables etc.

The 'Project Readiness Program' is aimed to make freshers comfortable with new technologies covering all the basic and topics. This also include business-oriented program which aims to give freshers professional touch of projects i.e. how projects are made and what are all the criteria to be covered in real time project

REQUIRED HARDWARE:

RAM:

1 GB

Hard Disk:

10 GB

Processor:

Intel Pentium 4 1.7 GHz

REQUIRED SOFTWARE:

Operating System:

Microsoft Windows 7 and above.

Database:

Relational Database Management System.

Tools:

SAP Logon Pad, Putty.

Language Requirement:

ABAP, C++

What is Linux?



From smartphones to cars, supercomputers and home appliances, the Linux operating system is everywhere.

Linux. It's been around since the mid '90s, and has since reached a user-base that spans industries and continents. For those in the know, you understand that Linux is actually everywhere. It's in your phones, in your cars, in your refrigerators, your Roku devices. It runs most of the Internet, the supercomputers making scientific breakthroughs, and the world's stock exchanges. But before Linux became the platform to run desktops, servers, and embedded systems across the globe, it was (and still is) one of the most reliable, secure, and worry-free operating systems available.

For those not in the know, worry not – here is all the information you need to get up to speed on the Linux platform.

What is Linux?

Just like Windows XP, Windows 7, Windows 8, and Mac OS X, Linux is an operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply – the operating system manages the communication between your software and your hardware. Without the operating system (often referred to as the “OS”), the software wouldn't function.

The OS is comprised of a number of pieces:

- **The Bootloader:** The software that manages the boot process of your computer. For most users, this will simply be a splash screen that pops up and eventually goes away to boot into the operating system.

- **The kernel:** This is the one piece of the whole that is actually called “Linux”. The kernel is the core of the system and manages the CPU, memory, and peripheral devices. The kernel is the “lowest” level of the OS.
- **Daemons:** These are background services (printing, sound, scheduling, etc) that either start up during boot, or after you log into the desktop.
- **The Shell:** You’ve probably heard mention of the Linux command line. This is the shell – a command process that allows you to control the computer via commands typed into a text interface. This is what, at one time, scared people away from Linux the most (assuming they had to learn a seemingly archaic command line structure to make Linux work). This is no longer the case. With modern desktop Linux, there is no need to ever touch the command line.
- **Graphical Server:** This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just “X”.
- **Desktop Environment:** This is the piece of the puzzle that the users actually interact with. There are many desktop environments to choose from (Unity, GNOME, Cinnamon, Enlightenment, KDE, XFCE, etc). Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, games, etc).
- **Applications:** Desktop environments do not offer the full array of apps. Just like Windows and Mac, Linux offers thousands upon thousands of high-quality software titles that can be easily found and installed. Most modern Linux distributions (more on this in a moment) include App Store-like tools that centralize and simplify application installation. For example: Ubuntu Linux has the Ubuntu Software Center (Figure 1) which allows you to quickly search among the thousands of apps and install them from one centralized location.



The Ubuntu software centre is a Linux app store that carries thousands of free and commercial applications for Linux.

Why use Linux?

This is the one question that most people ask. Why bother learning a completely different computing environment, when the operating system that ships with most desktops, laptops, and servers works just fine? To answer that question, I would pose another question. Does that operating system you're currently using *really* work "just fine"? Or are you constantly battling viruses, malware, slowdowns, crashes, costly repairs, and licensing fees?

If you struggle with the above and want to free yourself from the constant fear of losing data or having to take your computer in for the "yearly clean up," Linux might be the perfect platform for you. Linux has evolved into one of the most reliable computer ecosystems on the planet. Combine that reliability with zero cost of entry and you have the perfect solution for a desktop platform.

That's right, zero cost of entry...as in free. You can install Linux on as many computers as you like without paying a cent for software or server licensing (including costly Microsoft Client Access License – CALs).

Let's take a look at the cost of a Linux server, in comparison to Windows Server 2012. The price of the Windows Server 2012 software alone can run up to \$1,200.00 USD. That doesn't include CALs, and licenses for other software you may need to run (such as a database, a web server, mail server, etc). With the Linux server...it's all free and easy to install. In fact, installing a full-blown web server

(that includes a database server), is just a few clicks or commands away (take a look at “Easy LAMP Server Installation” to get an idea how simple it can be).

If you’re a system administrator, working with Linux is a dream come true. No more daily babysitting servers. In fact, Linux is as close to “set it and forget it” as you will ever find. And, on the off chance, one service on the server requires restarting, re-configuring, upgrading, etc... most likely the rest of the server won’t be affected.

Be it the desktop or a server, if zero cost isn’t enough to win you over – what about having an operating system that will work, trouble free, for as long as you use it? I’ve personally used Linux for nearly twenty years (as a desktop and server platform) and have not once had an issue with malware, viruses, or random computer slow-downs. It’s *that* stable. And server reboots? Only if the kernel is updated. It is not out of the ordinary for a Linux server to go years without being rebooted. That’s stability and dependability.

Linux is also distributed under an open source license. Open source follows the following key philosophies:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works and change it to make it do what you wish.
- The freedom to redistribute copies so you can help your neighbour.
- The freedom to distribute copies of your modified versions to others.

The above is crucial to understanding the community that comes together to create the Linux platform. It is, without a doubt, an operating system that is “by the people, for the people”. These philosophies are also one of the main reasons a large percentage of people use Linux. It’s about freedom and freedom of choice.

History of C++

The C++ programming language has a history going back to 1979, when Bjarne Stroustrup was doing work for his Ph.D. thesis. One of the languages Stroustrup had the opportunity to work with was a language called Simula, which as the name implies is a language primarily designed for simulations. The Simula 67 language - which was the variant that Stroustrup worked with - is regarded as the first language to support the object-oriented programming paradigm. Stroustrup found that this paradigm was very useful for software development, however the Simula language was far too slow for practical use.

Shortly thereafter, he began work on "C with Classes", which as the name implies was meant to be a superset of the C language. His goal was to add object-oriented programming into the C language, which was and still is a language well-respected for its portability without sacrificing speed or low-level functionality. His language included classes, basic inheritance, inlining, default function arguments, and strong type checking in addition to all the features of the C language.

The first C with Classes compiler was called Cfront, which was derived from a C compiler called CPre. It was a program designed to translate C with Classes code to ordinary C. A rather interesting point worth noting is that Cfront was written mostly in C with Classes, making it a self-hosting compiler (a compiler that can compile itself). Cfront would later be abandoned in 1993 after it became difficult to integrate new features into it, namely C++ exceptions. Nonetheless, Cfront made a huge impact on the implementations of future compilers and on the Unix operating system.

In 1983, the name of the language was changed from C with Classes to C++. The ++ operator in the C language is an operator for incrementing a variable, which gives some insight into how Stroustrup regarded the language. Many new features were added around this time, the most notable of which are virtual functions, function overloading, references with the & symbol, the const keyword, and single-line comments using two forward slashes (which is a feature taken from the language BCPL).

In 1985, Stroustrup's reference to the language entitled *The C++ Programming Language* was published. That same year, C++ was implemented as a commercial product. The language was not officially standardized yet, making the book a very important reference. The language was updated again in 1989 to include protected and static members, as well as inheritance from several classes.

In 1990, *The Annotated C++ Reference Manual* was released. The same year, Borland's Turbo C++ compiler would be released as a commercial product. Turbo C++ added a plethora of additional libraries which would have a considerable impact on C++'s development. Although Turbo C++'s last stable release was in 2006, the compiler is still widely used.

In 1998, the C++ standards committee published the first international standard for C++ ISO/IEC 14882:1998, which would be informally known as C++98. *The Annotated C++ Reference Manual* was said to be a large influence in the development of the standard. The Standard Template Library, which began its conceptual development in 1979, was also included. In 2003, the

committee responded to multiple problems that were reported with their 1998 standard and revised it accordingly. The changed language was dubbed C++03.

In 2005, the C++ standards committee released a technical report (dubbed TR1) detailing various features they were planning to add to the latest C++ standard. The new standard was informally dubbed C++0x as it was expected to be released sometime before the end of the first decade. Ironically, however, the new standard would not be released until mid-2011. Several technical reports were released up until then, and some compilers began adding experimental support for the new features.

In mid-2011, the new C++ standard (dubbed C++11) was finished. The Boost library project made a considerable impact on the new standard, and some of the new modules were derived directly from the corresponding Boost libraries. Some of the new features included regular expression support (details on regular expressions may be found [here](#)), a comprehensive randomization library, a new C++ time library, atomics support, a standard threading library (which up until 2011 both C and C++ were lacking), a new for loop syntax providing functionality similar to foreach loops in certain other languages, the auto keyword, new container classes, better support for unions and array-initialization lists, and variadic templates.

The Features of C++ as a Language

Now that all the necessary theory has been covered, now it is possible to explain what C++ has to offer as a programming language.

C++

- ...is an open ISO-standardized language.
For a time, C++ had no official standard and was maintained by a de-facto standard, however since 1998, C++ is standardized by a committee of the ISO. Their page may be accessed [here](#).
- ...is a compiled language.
C++ compiles directly to a machine's native code, allowing it to be one of the fastest languages in the world, if optimized.
- ...is a strongly-typed unsafe language.
C++ is a language that expects the programmer to know what he or she is doing, but allows for incredible amounts of control as a result.
- ...supports both manifest and inferred typing.
As of the latest C++ standard, C++ supports both manifest and inferred typing, allowing flexibility and a means of avoiding verbosity where desired.
- ...supports both static and dynamic type checking.
C++ allows type conversions to be checked either at compile-time or at run-time, again offering another degree of flexibility. Most C++ type checking is, however, static.
- ...offers many paradigm choices.
C++ offers remarkable support for procedural, generic, and object-oriented programming paradigms, with many other paradigms being possible as well.
- ...is portable.
As one of the most frequently used languages in the world and as an open language, C++ has a wide range of compilers that run on many different platforms that support it. Code that exclusively uses C++'s standard library will run on many platforms with few to no changes.
- ...is upwards compatible with C
C++, being a language that directly builds off C, is compatible with almost all C code. C++ can use C libraries with few to no modifications of the libraries' code.
- ...has incredible library support.
A search for "library" on the popular project-management website SourceForge will yield over 3000 results for C++ libraries. A link to the results of the search may be found [here](#).

Logic Building on PBLApp by Wipro

Objective of Day8: Learn to solve number based problems (will require loops)

Question Title	Mettl Practice Test Link
digitSum : sum of all digits in N	https://tests.mettl.com/authenticateKey/ab1d60cc
digitSum even: sum of even digits in N	https://tests.mettl.com/authenticateKey/b55d1714
digitSum odd: sum of odd digits in N	https://tests.mettl.com/authenticateKey/738fdee0
digitSum opt: sum of even or odd digits	https://tests.mettl.com/authenticateKey/a05abbcf

Objective of Day9: Learn to solve number based problems

Question Title	Mettl Practice Test Link
Is Palindrome Number?	https://tests.mettl.com/authenticateKey/28c41d9d
Is Palindrome Possible?	https://tests.mettl.com/authenticateKey/f4fdb02
Create PIN using alpha, beta, gamma	https://tests.mettl.com/authenticateKey/be582d9f
Weight of a hill pattern	https://tests.mettl.com/authenticateKey/d612c0e6

Objective of Day10: Learn to solve number & string based problems

Question Title	Mettl Practice Test Link
Return second word in Uppercase	https://tests.mettl.com/authenticateKey/4a72723f
is Palindrome (string)	https://tests.mettl.com/authenticateKey/ffe8042
weight of string	https://tests.mettl.com/authenticateKey/387952fc
Most Frequent Digit	https://tests.mettl.com/authenticateKey/916310b8

Solve the below questions on mettl

Note - These are same questions as day-3 questions. The focus on day-4 is to learn the usage of mettl.

Question Title	Mettl Practice Test Link
Is Even?	https://tests.mettl.com/authenticateKey/2bd025dc
Is Odd?	https://tests.mettl.com/authenticateKey/dbdac2a9
Return last digit of the given number	https://tests.mettl.com/authenticateKey/454f012b
Return second last digit of given number	https://tests.mettl.com/authenticateKey/9f87004e
Sum of last digits of two given numbers	https://tests.mettl.com/authenticateKey/783a1fcf

Objective of Day5: Learn the use of division / and mod % operations to solve problems

Question Title	Mettl Practice Test Link
Is N an exact multiple of M?	https://tests.mettl.com/authenticateKey/36c4ef58
Of given 5 numbers, how many are even?	https://tests.mettl.com/authenticateKey/8edbe922
Of given 5 numbers, how many are odd?	https://tests.mettl.com/authenticateKey/67147bd5
Of 5 numbers, how many are even or odd?	https://tests.mettl.com/authenticateKey/607636d7

Logic building approach: Read the document "An Approach to Logic Building" (mainly section-B) available in the "Logic Building" link in PBLApp. It suggests a "Divide-and-Conquer" approach to be followed by beginners while developing logic. Students are expected to read the document carefully and follow the approach while solving all subsequent logic building problems.

Objective of Day6: Learn to solve math based number problems (will require loops and maths)

Question Title	Mettl Practice Test Link
Is Prime?	https://tests.mettl.com/authenticateKey/b1efaa3d

NOTE: Students are encouraged to think of various possible solutions, Write the algorithm (steps) on a paper, and then try it in mettl.

String based programs (part-1) -

Question Title	Mettl Practice Test Link
FindStringCode	https://tests.mettl.com/authenticateKey/e4df74e5
Get Code Through Strings	https://tests.mettl.com/authenticateKey/75c9faf0
Addition using Strings	https://tests.mettl.com/authenticateKey/2f174a80

Array based programs -

Question Title	Mettl Practice Test Link
Simple Encoded Array	https://tests.mettl.com/authenticateKey/1557c062
Decreasing sequence	https://tests.mettl.com/authenticateKey/8e68e9a4
Most Frequently Occurring Digit	https://tests.mettl.com/authenticateKey/cbe4c4da

Number based programs -

Question Title	Mettl Practice Test Link
Sum of Powers of Digits	https://tests.mettl.com/authenticateKey/92437794
Sum of Sums of Digits in Cyclic order	https://tests.mettl.com/authenticateKey/1ddb65e

String based programs (part-2) -

Question Title	Mettl Practice Test Link
Identify possible words	https://tests.mettl.com/authenticateKey/13486c16
Encoding Three Strings	https://tests.mettl.com/authenticateKey/f05028d5

What are Logic Building Hours (LBHs)?
Logic Building Hours (LBHs) are dedicated hours per day, when the students are expected to work on logic building exercises.

The previous two series of LBHs should hopefully have helped you to develop your logic building skills. Here is yet another edition of the LBHs Plan.

Number based programs -

Question Title	Mettl Practice Test Link
Generate series and find Nth element	https://tests.mettl.com/authenticateKey/d2272e10
Find result after alternate add_sub on N	https://tests.mettl.com/authenticateKey/814effc3
Find Password (stable unstable)	https://tests.mettl.com/authenticateKey/5106dfd
Calculate sum of non-prime index values	https://tests.mettl.com/authenticateKey/596e522f
Find the one digit to be removed to form palindrome	https://tests.mettl.com/authenticateKey/6d825776

String based programs -

Question Title	Mettl Practice Test Link
The "Nambiar Number" Generator	https://tests.mettl.com/authenticateKey/7db6c8a4
User ID Generation	https://tests.mettl.com/authenticateKey/592740f3
Message controlled Robot movement	https://tests.mettl.com/authenticateKey/aedcc3a6

Happy Coding!!

Codes

```
#include<iostream>

using namespace std;

class Date
{
public:
    Date() : day(0), mth(0), year(0) { }
    Date(int d, int m, int y) : day(d), mth(m), year(y) { }
    Date operator +(int n)
    {
        Date temp;
        temp.day=day+n;
        if(temp.day>=31)
        {
            temp.day=temp.day-31;
            temp.mth=mth+1;
        }
        else
            temp.mth=mth;
        if(temp.mth>=12)
        {
            temp.year=year+1;
            temp.mth=temp.mth-12;
        }
        else
            temp.year=year;
        return (temp);
    }
};
```

```

}
void print()
{
    cout<<"Day is "<<day<<endl;
    cout<<"Month is "<<mth<<endl;
    cout<<"Year is "<<year<<endl;
}
Date operator -(int n)
{
    Date temp;
    temp.day=day-n;
    if(temp.day<=0)
    {
        temp.day=temp.day+31;
        temp.mth=mth-1;
    }
    else
    temp.mth=mth;
    if(temp.mth<=0)
    {
        if(temp.mth==0)
        {
            temp.mth=temp.mth+1;
        }
        else
        {
            temp.year=year-1;
            temp.mth=temp.mth+13;

```

```

        }
    }
    else
        temp.year=year;
        return (temp);
    }
private:
    int day;
    int mth;
    int year;
};

int main()
{
    Date d1(17,8,2017);
    Date d2;
    d2=d1+5;
    d2.print();
    d2=d1-5;
    d2.print();
}

```

What is SAP?

SAP stands for **S**ystems **A**pplications and **P**roducts in Data Processing.

SAP by definition is also named of the ERP (Enterprise Resource Planning) software as well the name of the company.

SAP Software was Founded in 1972 by Wellenreuther, Hopp, Hector, Plattner and Tschira.

SAP system consists of a number of fully integrated modules, which covers virtually every aspect of the business management.

SAP is #1 in the ERP market. As of 2010, SAP has more than 140,000 installations worldwide, over 25 industry-specific business solutions and more than 75,000 customers in 120 countries

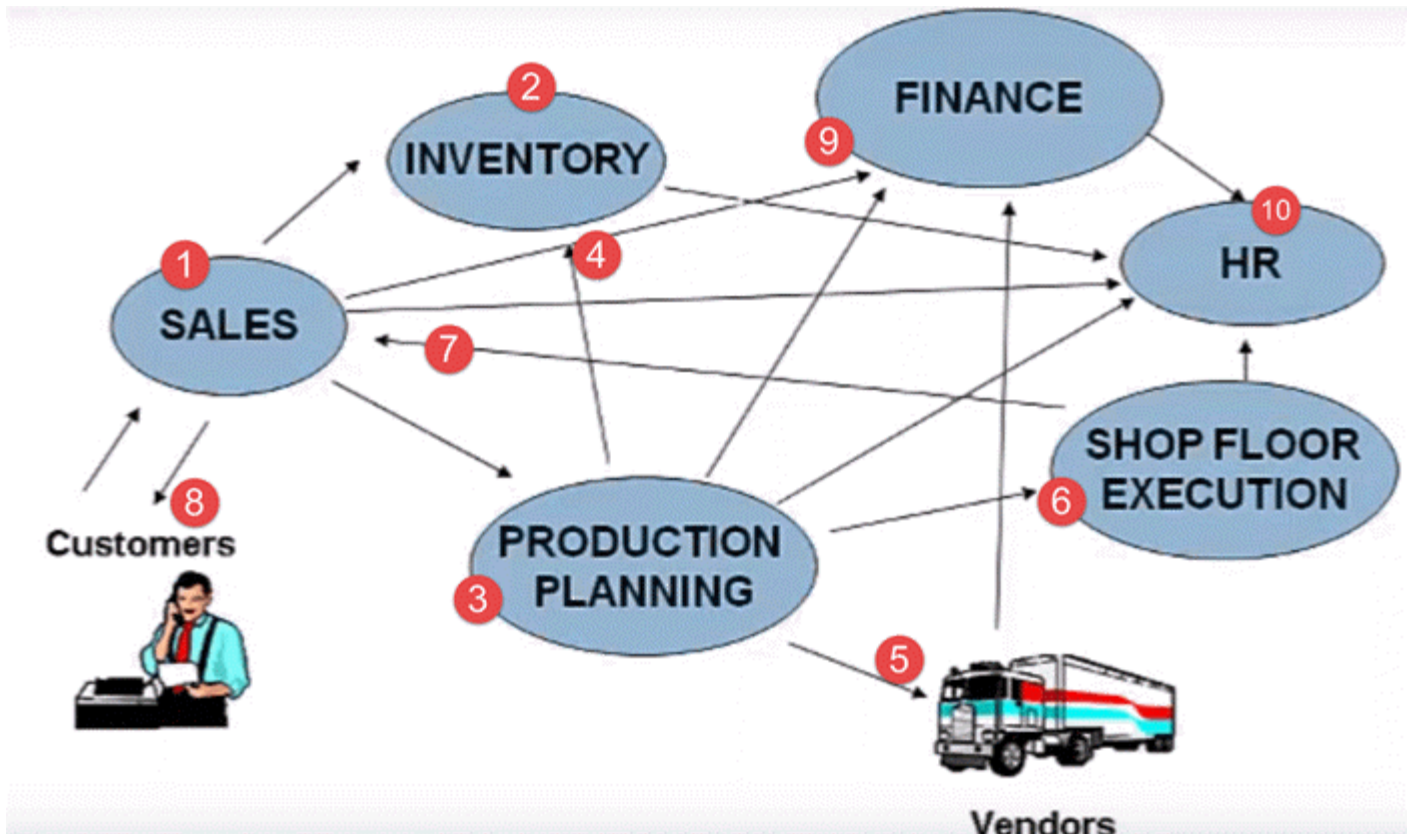
Other Competitive products of SAP Software in the market are Oracle, Microsoft Dynamics etc.

What is SAP ERP? Why it is Required?

The very basic question to any beginners is why Enterprise Resource Planning also called ERP is required? To answer this, let's examine this typical business scenario.

Suppose a client approaches sales team asking for a particular product. The sales team contacts to inventory department to check the availability of the product. To their surprise, sales team found out that the product is out of stock. So next time this don't happen, they have to introduce a SAP ERP tool.

Before we actually see in detail, what ERP is and how ERP can help in your business process, we will understand how different departments are involved in the whole business process, right from the ordering of the raw material – to manufacturing goods – to delivering final goods to the customer.



Here is the whole process that is followed by any business unit.

1. Client contacts the sales team to check the availability of the product
2. Sales team approaches the Inventory department to check for the availability of the product
3. In case the product is out of stock, the sales team approaches the Production Planning Department to manufacture the product
4. The production planning team checks with inventory department for availability of raw material
5. If raw material is not available with inventory, the Production Planning team buys the raw material from the Vendors
6. Then Production Planning forwards the raw materials to the Shop Floor Execution for actual production
7. Once ready, the Shop Floor Team forwards the goods to the Sales Team
8. Sales Team who in turn deliver it to the client
9. The sales team updates the finance with revenue generated by the sale of the product. Production planning team update the finance with payments to be made to different vendors for raw materials.
10. All departments approach the HR for any Human Resource related issue.

That is a typical business process for any manufacturing company. Some key inferences one could derive from the scenario would be.

- It has many departments or business units

- These departments or business units continuously communicate and exchange data with each other
- The success of any organization lies in effective communication, and data exchange, within these departments, as well as associated third party such as vendors, outsourcers, and customers.

Based on the manner in which communication and data exchanged is managed. Enterprise systems can be broadly classified as

1) Decentralized System

2) Centralized System which are also called as ERP.

Decentralized System

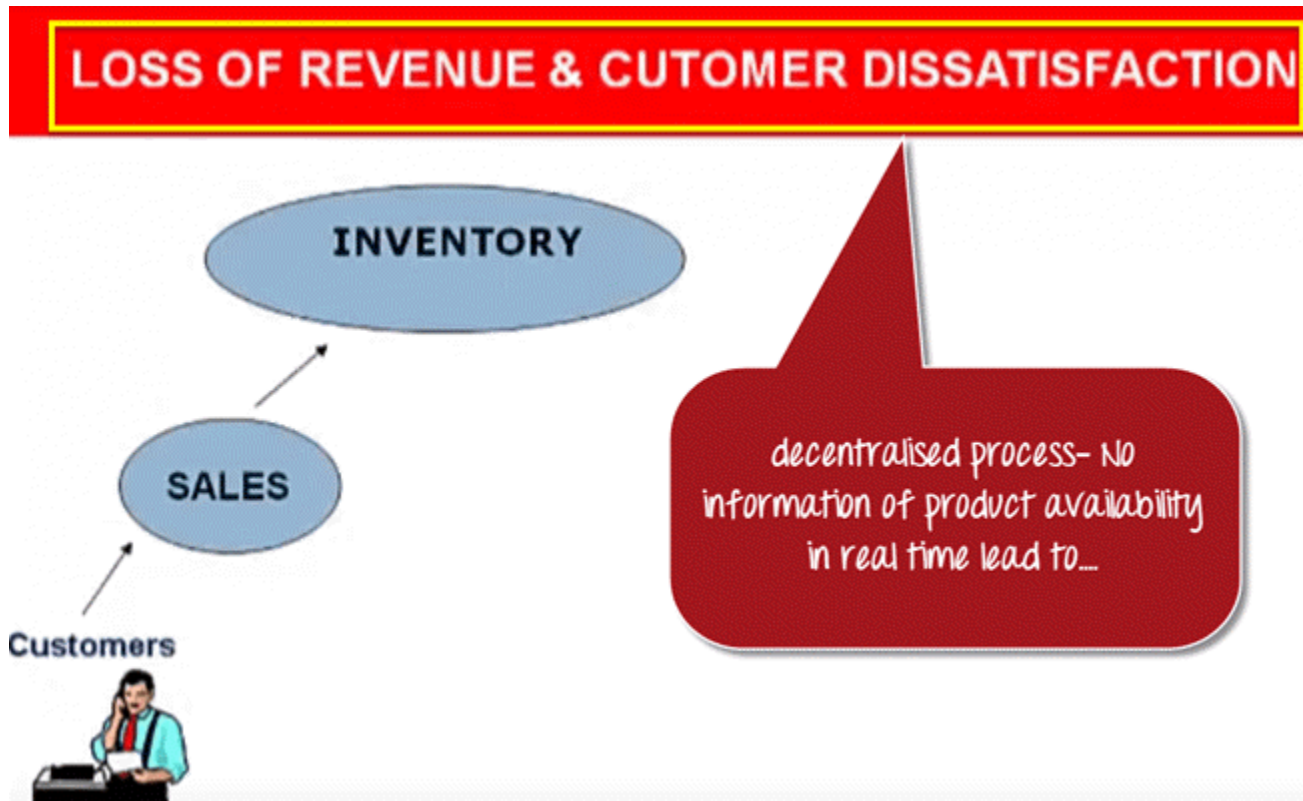
Let's look at Decentralized system first, in a company with Decentralized System of Data Management, there are two major problems –

- Data is maintained locally at the individual departments
- Departments do not have access to information or data of other departments

To identify problems arising due to decentralized Enterprise management system lets look at the same business process again. The customer approaches the sales team for a product, but this time around he needs the product, on an urgent basis.



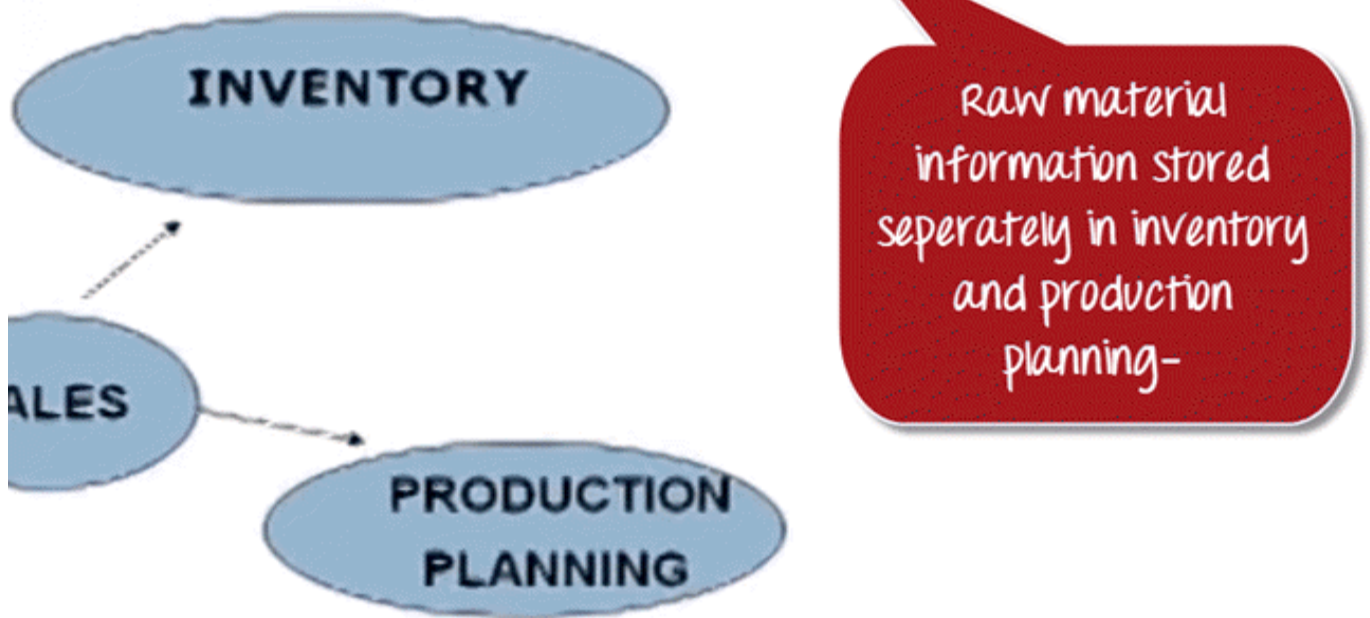
Since it is a decentralized process, the Sales Team do not have any real-time information access to the products availability. So they approach the Inventory department to check the availability of the product. This process takes time and customer chooses another vendor leading to loss of revenue and customer dissatisfaction.



Now, suppose the product is out of stock and the Sales Team approaches the Production Planning team to manufacture the product for future use. Production Planning Team checks the availability of the raw materials required.

In a decentralized system, raw material information is separately stored by Production Planning as well as Inventory Department. Thus, data maintenance cost (in this case Raw Material) goes up.

DATA MAINTAINENCE COST GOES UP



The raw material information is available in two different departments Inventory as well as Production Planning. When sales team check a particular raw material required to manufacture the product, it shows the raw material is available as per the inventory, but as per the database of the production planning team, the raw material is out of stock.

So, they go ahead and buy the raw material. Thus, material as well inventory cost goes up.

Once the raw material is available, the shop floor department suddenly realizes they are short of workers they approach the HR, who in turn hire temporary employees at higher than market rates. Thus LABOR Cost Increases.

The production planning department fails to update the finance department on the materials they have purchased. The finance department defaults the payment deadline set by the vendor causing the company loss of its reputation and even inviting a possible legal action.

LABOUR COST INCREASES

unplanned work schedule in a decentralized system result in hiring temporary employees higher than market rate

HR

These are just a few of many problems with decentralized systems.

Some Major problems with the decentralized system are –

- Numerous disparate information system generates individually over time which are difficult to maintain
- Integrating the data is time and money consuming
- Inconsistencies and duplication of data
- Lack of timely information leads to customer dissatisfaction , loss of revenue and reputation
- High Inventory, material, and human resource cost.

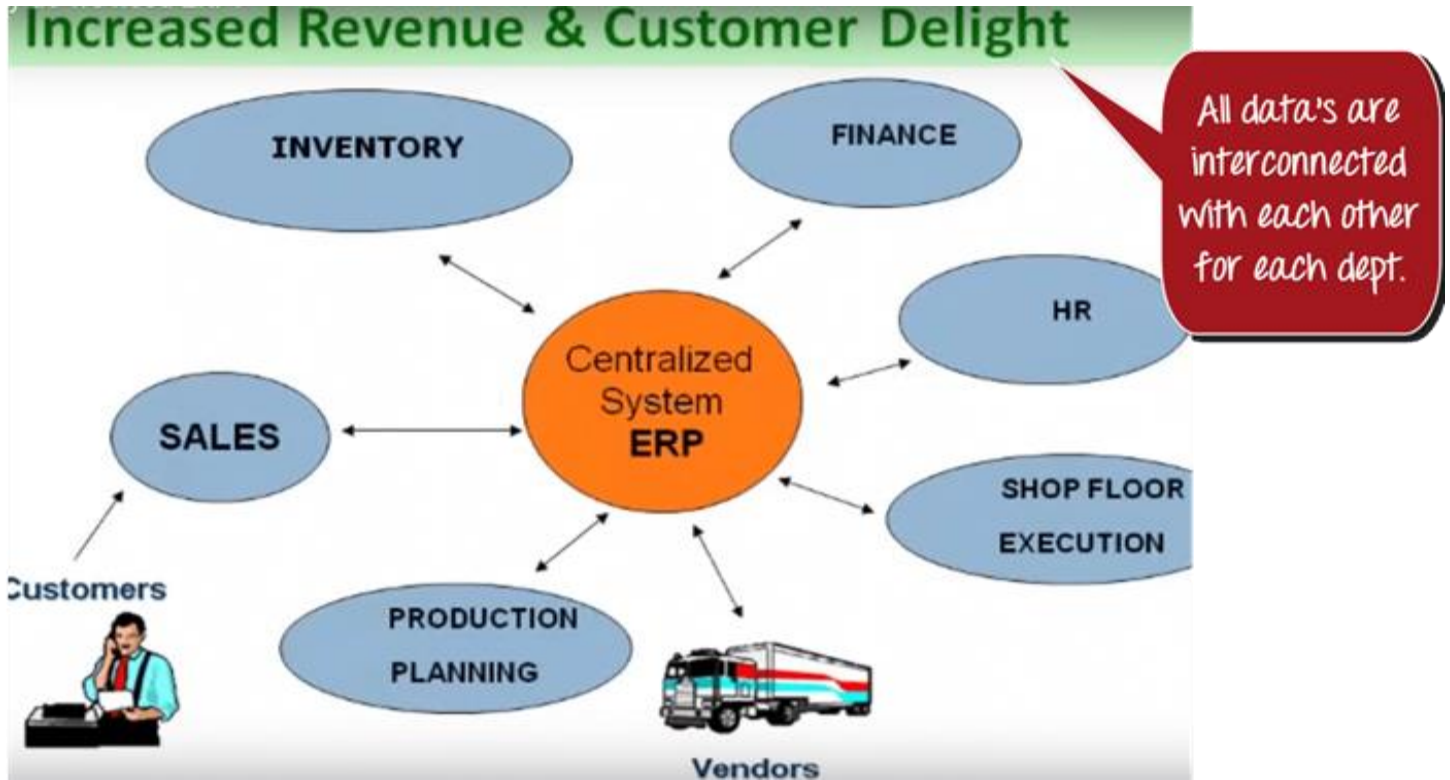
These are some major drawbacks for which we need a solution. Well the Solution lies in Centralized Systems i.e. ERP.

Centralized System

In a company, with Centralized System of Information and Data Management.

- 1) Data is maintained at a central location and is shared with various Departments
- 2) Departments have access to information or data of other Departments

Let's look at the same business process again to understand how a Centralized Enterprise System helps to overcome problems posed by a Decentralized Enterprise System.



In this Case, all departments update a Central Information System.

- When Customer approaches the sales team to buy a product on an urgent basis. The Sales Team has real-time information access to the products in inventory which is updated by the Inventory Department in the Centralized System
- Sales Team respond to customer request on time leading to Increased Revenue and Customer Delight.
- In case, manufacturing is required the Sales Team update the Centralized Database, so that all the department remain informed about the product status.
- Production Planning Department is auto updated by the Centralized Database for requirements. Production Planning Team checks the availability of the raw materials required via Central Database, which is updated by the Inventory Department.
- Thus, Data Duplication is avoided, and accurate data is made available. The Shop Floor Team update their Man Power Status regularly in the Central Database, which can be accessed by the HR department.
- In case of shortage of workforce, HR team starts recruitment process with considerable lead time to hire a suitable candidate at market price. Thus labor cost goes down.
- While vendors can directly submit their invoices to the Central Enterprise System, which can be accessed by the finance department. Thus, payments are made on time, and possible legal actions are avoided
- SAP software is a type of Centralized System. SAP System is most popularly used ERP software.

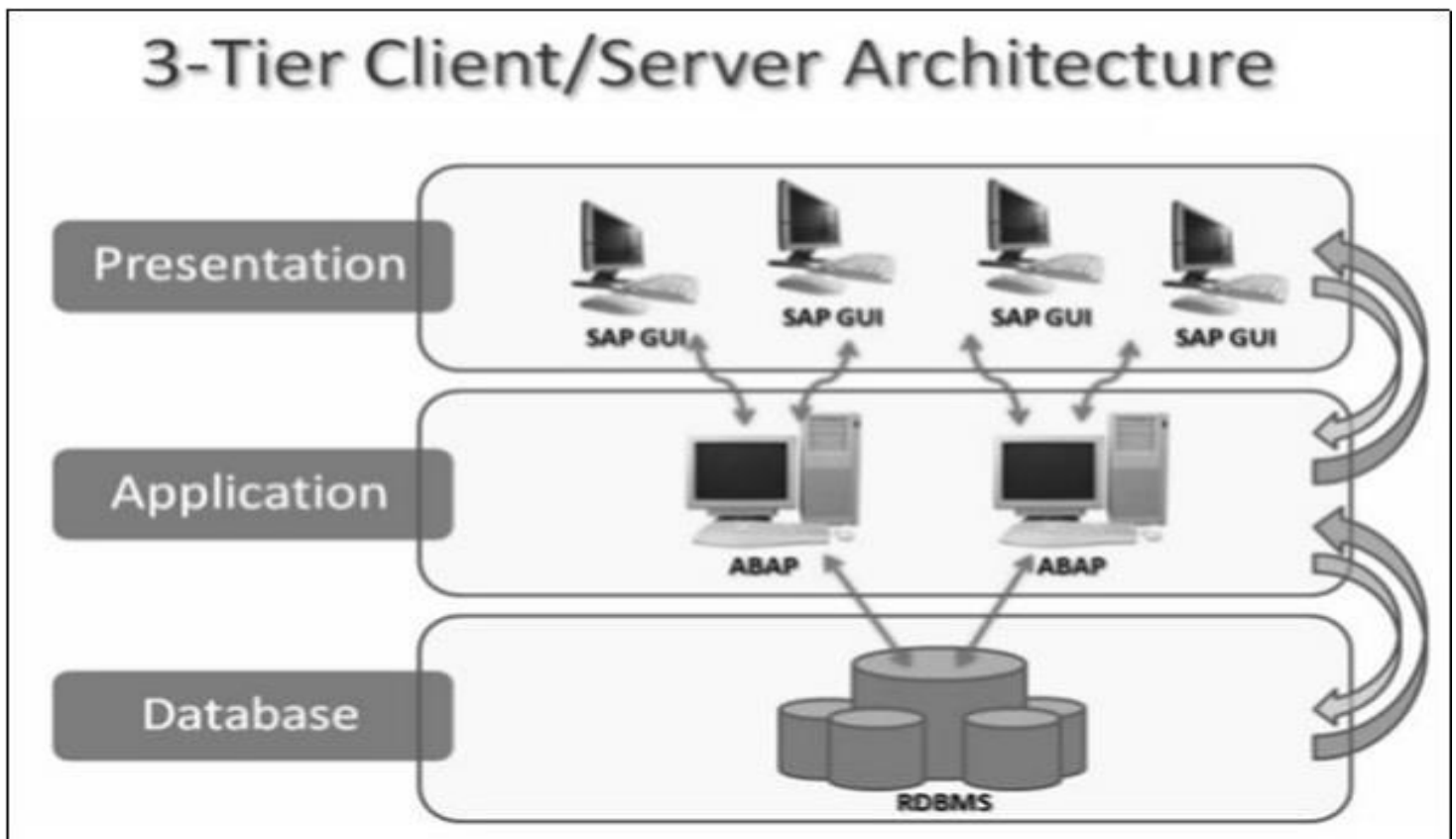
Key benefits of the centralized system are:

- It Eliminates the duplication, discontinuity and redundancy in data
- Provides information across departments in real time.
- SAP System is Provides control over various business processes
- Increases productivity, better inventory management , promotes quality , reduced material cost, effective human resources management, reduced overheads boosts profits
- Better customer interaction and increased throughput. It also improves customer service
- Hence, a centralized enterprise management system is required.
- SAP Software is a centralized enterprise management system also known as Enterprise Resource Planning.

Types of SAP Consultants

- Functional Consultant - They are responsible for customizing SAP as per customer demand. They talk with developers to code custom ABAP programs as per client requirements.
- Technical Consultant - They are responsible for coding ABAP.

SAP IS A 3 TIER SYSTEM



ABAP(ADVANCED BUSINESS APPLICATION PROGRAMING)

ABAP (Advanced Business Application Programming) is the primary programming language supported on the SAP NetWeaver ABAP application server platform and applications that run on it, such as SAP ERP (formerly R/3), S/4HANA and CRM. SAP uses ABAP to implement its own applications on the NetWeaver ABAP platform, and SAP customers use ABAP to modify the functionality of SAP applications or build their own on the NetWeaver ABAP platform. ABAP is the oldest and, likely, the most widely used of SAP's four major application platforms, which also includes SAP NetWeaver Java, SAP HANA and SAP Cloud Platform.

Evolution of ABAP

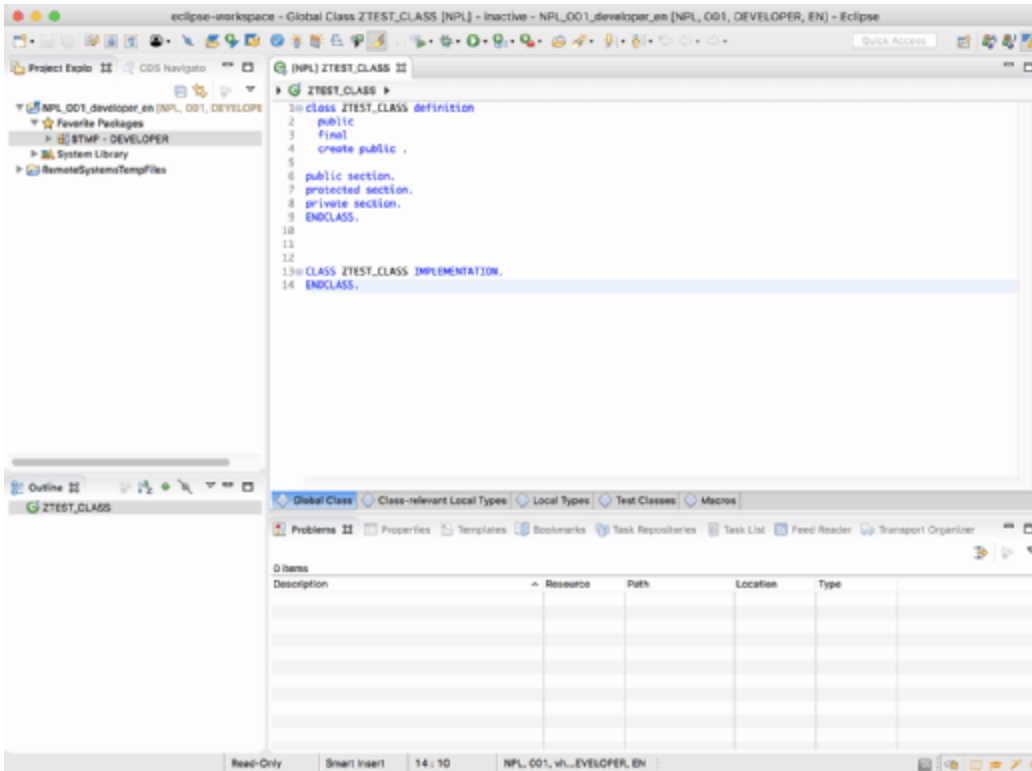
SAP ABAP began in the 1980s as a report-generation language in SAP products. It took on a central role in SAP R/3 as the enterprise resource planning (ERP) system's primary implementation and extension language. Over the years, it gained new features, most notably the introduction of object-oriented constructs, referred to as ABAP Objects, in 1999 and the introduction of new database access methods and a large amount of new syntax starting around 2010.

ABAP features are tightly coupled with the SAP R/3 or NetWeaver release that is being used. The only way to access new features of the language is to upgrade to a newer release of the ABAP application server. In many cases, programs written using features of a newer application server version will not run on older SAP systems.

ABAP development tools

By far the largest developer of ABAP code is SAP itself. However, many thousands of ABAP developers work with SAP customers and consulting companies to maintain and modify SAP systems. ABAP is regularly in the top 30 of the Tiobe Index, which roughly tracks the popularity of programming languages.

Developers who work in ABAP usually do so either in the ABAP Development Tools (a set of plugins for the Eclipse open source Java development platform) or in the ABAP Workbench transaction in the SAP graphical user interface (GUI). Both environments offer a set of tools to assist development, from code completion to automated testing tools.



ETHAN JEWETT

An empty ABAP Class in the ABAP Developer tools as seen in Eclipse.

SAP Solution Manager also offers tools for managing the development lifecycle of ABAP code. There is little support for development tooling beyond what SAP provides, though some customers have built their own integrations with third-party continuous integration, version control and bug tracking tools.

Special features and the larger ABAP infrastructure

ABAP doesn't stand alone, and it is highly integrated with other features of the SAP NetWeaver ABAP application server. Among these are the following:

- **Logical database connections**, which allow code to be abstracted from a specific database. The actual database connections are configured outside of ABAP code, allowing the same code to be used in different database environments.
- **Open SQL**, an abstraction of SQL syntax that is part of the ABAP language and which the ABAP runtime environment converts to native SQL that is appropriate for the database being used. Open SQL has many similarities to Microsoft .NET's Language Integrated Query (LINQ) concept.

- **Internal Tables**, which hold collections of objects that are accessed using special language keywords or Open SQL. This ABAP concept contrasts with the concept of typed arrays like in Java or C++.
- **Security**, in which ABAP is integrated with SAP NetWeaver's security infrastructure.
- **Data Dictionary**, a universal dictionary of data structure definitions, often including business logic, which is available to all ABAP programs in a system.
- **Change and Transport System (CTS)**, which tracks changes to development objects and manages the promotion of development objects to quality assurance and production environments.
- **Shared development system**, which is an important aspect of ABAP. ABAP differs from most newer languages in that development usually takes place on a shared system, with all developers working on the same set of development objects at the same time.

Code

REPORT ztest_display NO STANDARD PAGE HEADING.

DATA: wa_zchitkara_staff TYPE zchitkara_staff,
 wa_zchitkara_studen TYPE zchitkara_studen,
 it_zchitkara_studen TYPE TABLE OF zchitkara_studen.

*DATA: d_studid TYPE z_student_id,

* d_mobno TYPE z_mobno,

* d_fname TYPE z_fname,

* d_mname TYPE z_mname,

* d_lname TYPE z_lname,

* d_sect TYPE z_section,

DATA sep TYPE c VALUE ''.

DATA : it_detail TYPE TABLE OF zstudent_detail,
 wa_detail TYPE zstudent_detail.

SELECTION-SCREEN: BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.

PARAMETERS: d_sid TYPE z_sid OBLIGATORY.

SELECTION-SCREEN: END OF BLOCK b1.

SELECTION-SCREEN: BEGIN OF BLOCK b2 WITH FRAME TITLE text-002.

PARAMETERS: a TYPE char20,

- b type char10,
 c TYPE char10.

SELECTION-SCREEN: END OF BLOCK b2.

SELECTION-SCREEN: BEGIN OF BLOCK b3 WITH FRAME TITLE text-003.

PARAMETERS: d TYPE char20,

 e TYPE char10,

* f type char10,

 g TYPE char10.

SELECTION-SCREEN: END OF BLOCK b3.

AT SELECTION-SCREEN OUTPUT.

 LOOP AT SCREEN.

 IF screen-name = 'A'.


```

    screen-input = 0.
    MODIFY SCREEN.
ENDIF.
ENDLOOP.
LOOP AT SCREEN.
    IF screen-name = 'C' .
        screen-input = 0.
        MODIFY SCREEN.
    ENDIF.
ENDLOOP.

```

- LOOP AT SCREEN.


```

                IF screen-name = 'D' .
                    screen-input = 0.
                    MODIFY SCREEN.
                ENDIF.
            ENDLOOP.

```

```

LOOP AT SCREEN.
    IF screen-name = 'E' .
        screen-input = 0.
        MODIFY SCREEN.
    ENDIF.
ENDLOOP.

```

```

LOOP AT SCREEN.
    IF screen-name = 'G' .
        screen-input = 0.
        MODIFY SCREEN.
    ENDIF.
ENDLOOP.

```

AT SELECTION-SCREEN.

```

* wa_zchitkara_studen-z_sid = d_sid.
* wa_zchitkara_staff-z_sid = d_sid.
* wa_zchitkara_staff-z_fname = d_fname.
* wa_zchitkara_staff-z_lname = d_lname.
* wa_zchitkara_studen-z_fname = d_fname.
* wa_zchitkara_studen-z_lname = d_lname.
* wa_zchitkara_staff-z_mobno = d_mobno.
* wa_zchitkara_studen-z_mobno = d_mobno.
* wa_zchitkara_studen-z_section = d_sect.

```

- * *SELECT SINGLE* *
- * *FROM* zchitkara_staff
- * *INTO* wa_zchitkara_staff
- * *WHERE* z_sid = d_sid.

CALL FUNCTION 'ZSTAFF_DETAILS'
 EXPORTING
 z_sid = d_sid
 IMPORTING
 ls_zchitkara_staff = wa_zchitkara_staff.

IF sy-subrc = 0.

- * *WRITE:/ 'Mentor'.*
- * *ULINE.*

CONCATENATE wa_zchitkara_staff-z_fname wa_zchitkara_staff-
 z_lname INTO a SEPARATED BY sep.

- * *a = wa_zchitkara_staff-z_fname.*
- * *b = wa_zchitkara_staff-z_lname.*
- c = wa_zchitkara_staff-z_mobno.*

- * *WRITE:/ wa_zchitkara_staff-z_fname ,sy-vline,*
 - * *wa_zchitkara_staff-z_lname,sy-vline,*
 - * *wa_zchitkara_staff-z_mobno.*
 - * *ULINE.*
- ENDIF.

- * *SELECT* *
- * *FROM* zchitkara_studen
- * *INTO TABLE* it_zchitkara_studen
- * *WHERE* z_sid = d_sid

CALL FUNCTION 'ZSTUDENT_DETAILS'
 EXPORTING
 z_sid = d_sid
 TABLES
 lt_zchitkara_studen = it_zchitkara_studen.

IF it_zchitkara_studen IS NOT INITIAL.

 LOOP AT it_zchitkara_studen INTO wa_zchitkara_studen.

```

*   WRITE:/ 'Student'.
*   ULINE.
    CONCATENATE wa_zchitkara_studen-z_fname wa_zchitkara_studen-
z_lname INTO e SEPARATED BY sep.
    d = wa_zchitkara_studen-z_student_id.
*   e = wa_zchitkara_studen-z_fname.
*   f = wa_zchitkara_studen-z_lname.
    g = wa_zchitkara_studen-z_mobno.
*   WRITE:/ wa_zchitkara_studen-z_student_id,sy-vline,
*           wa_zchitkara_studen-z_fname,sy-vline,
*           wa_zchitkara_studen-z_lname,sy-vline,
*           wa_zchitkara_studen-z_mobno.
*   ULINE.

```

```

wa_detail-z_sid = d_sid.
wa_detail-z_mobno = wa_zchitkara_staff-z_mobno.
wa_detail-z_student_id = wa_zchitkara_studen-z_student_id.
wa_detail-z_fmobno = wa_zchitkara_studen-z_fmobno.
APPEND wa_detail TO it_detail.
ENDLOOP.

```

- START-OF-SELECTION.
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
EXPORTING
i_structure_name = 'ZSTUDENT_DETAIL'
i_default = 'X'
i_save = 'A'
TABLES
t_outtab = it_detail.

References

1. <https://www.linux.com/what-is-linux>
2. <http://www.cplusplus.com/info/>
3. <http://www.cplusplus.com/info/description/>
4. <http://www.cplusplus.com/info/history/>
5. <https://www.guru99.com/what-is-sap-definition-of-sap-erp-software.html>
6. <https://searchsap.techtarget.com/definition/ABAP>