

INTERNATIONAL SCHOOL OF MANAGEMENT & TECHNOLOGY

**ASSIGNMENT COVER SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| STUDENT DETAILS | | | |
|  | | | |
| Student ID |  | **Reg No.** |  |
| Family Name | DEVKOTA | **Given Name** | ANISH |
| Enrolment Year | 2020 | **Section** | L |
| Semester | FIRST | **Email** | anishdevkota@ismt.edu.np |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UNIT DETAILS | | | | |
| Unit Title | **Programming** | **Unit Code** | **D/615/1618** | |
| Assessor Name | **Anoj Shrestha** | **Issued Date** | **29th march 2021** | |
| Assignment Title | **Inventory System** | | | |
| Assignment No | **1/1** | **Submission Date** | | **21st june 2021** |
| Qualification |  | **Campus** | | **ISMT** |





**STUDENT ASSESSMENT SUBMISSION AND DECLARATION**

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name | Anish devkota | Assessor Name | Anoj Shrestha |
| Issue Date | 29th march 2021 | Submission Date | 21st june 2021 |
| Programme | BTECH HND leading to bsc.IT | | |
| Unit Name | Programming | | |
| Assignment Title | Inventory System | | |

**Plagiarism**

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalized. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

**Student Declaration**

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: Anish Devkota Date: june 21st 2021



Pearson Education 2018  
Higher Education Qualifications

Contents

[**Section A** 6](#_Toc73357483)

[**Report** 6](#_Toc73357484)

[**Introduction to Algorithm** 7](#_Toc73357485)

[**Characteristics of Algorithm** 7](#_Toc73357486)

[**Advantages:** 7](#_Toc73357487)

[**Disadvantages**: 7](#_Toc73357488)

[**Examples of Algorithm with flowchart and its working** 7](#_Toc73357489)

[**Outline process of developing application** 9](#_Toc73357490)

[**Development environment** 10](#_Toc73357491)

[**Programming Language** 10](#_Toc73357492)

[**Platform environment** 11](#_Toc73357493)

[**Inventory management system design layout** 11](#_Toc73357494)

[**Design forms of inventory system created in visual studio 19:** 13](#_Toc73357495)

[**Data dictionary:** 15](#_Toc73357496)

[**Algorithm and flowchart of application** 16](#_Toc73357497)

[**Code execution process** 20](#_Toc73357498)

[**Implementation of algorithm** 21](#_Toc73357499)

[**Relation between algorithm and source code**: 23](#_Toc73357500)

[**Section B** 23](#_Toc73357501)

[**Introduction to Procedural Oriented language** 23](#_Toc73357502)

[**Features of Procedural Oriented Language** 23](#_Toc73357503)

[**Limitations of Procedural Oriented Language** 24](#_Toc73357504)

[**Function of Procedural Oriented programming** 24](#_Toc73357505)

[**Basic** I/O 24](#_Toc73357506)

[**Control Statement** 24](#_Toc73357507)

[**Looping** 28](#_Toc73357508)

[**Array** 30](#_Toc73357509)

[**Structure** 31](#_Toc73357510)

[**Function** 32](#_Toc73357511)

[**Pointer** 33](#_Toc73357512)

[**String** 33](#_Toc73357513)

[**File** **handling** 33](#_Toc73357514)

[**Introduction to Object Oriented Language** 34](#_Toc73357515)

[**Feature of OOPL** 34](#_Toc73357516)

[**Classes** 34](#_Toc73357517)

[**Object** 34](#_Toc73357518)

[**Data encapsulation** 35](#_Toc73357519)

[**Inheritance** 35](#_Toc73357520)

[**Polymorphism** 37](#_Toc73357521)

[**Event driven programming language** 38](#_Toc73357522)

[**Characteristics/Features of event driven program** 38](#_Toc73357523)

[**Advantages of Event driven programming language** 39](#_Toc73357524)

[**Disadvantages of event driven programming language** 39](#_Toc73357525)

[**Relationship between three programming language** 39](#_Toc73357526)

[**Critical evaluation of source code of application** 40](#_Toc73357527)

[**Justification** 42](#_Toc73357528)

[**Introduction to IDE** 43](#_Toc73357529)

[**Benefits of IDE** 43](#_Toc73357530)

[**Tools available in IDE** 43](#_Toc73357531)

[**Conclusion:** 45](#_Toc73357532)

[**Implementation of Application** 45](#_Toc73357533)

[**Design and code used for inventory system** 45](#_Toc73357534)

[**IDE to manage the development process** 78](#_Toc73357535)

[**Code editor** 78](#_Toc73357536)

[**Compiler**: 79](#_Toc73357537)

[**Documentation:** 79](#_Toc73357538)

[**Libraries**: 80](#_Toc73357539)

[**Build automatic:** 80](#_Toc73357540)

[**Debug**: 80](#_Toc73357541)

[**Constraints of visual studio:** 80](#_Toc73357542)

[**Conclusion** 80](#_Toc73357543)

[**Debugging** 80](#_Toc73357544)

[**Process of debugging** 80](#_Toc73357545)

[**Debugging facilities of IDE** 81](#_Toc73357546)

[**Debugging process evaluation:** 82](#_Toc73357547)

[**Coding standard** 83](#_Toc73357548)

[**Benefits of code standard:** 83](#_Toc73357549)

[**Constraints of code standard:** 83](#_Toc73357550)

[**Coding standard followed while delivering our application** 84](#_Toc73357551)

[**Indentation** 84](#_Toc73357552)

[**Commenting code** 84](#_Toc73357553)

[**Declaration of variable** 84](#_Toc73357554)

[**Naming convention** 85](#_Toc73357555)

[**Pascal naming convention** 85](#_Toc73357556)

[**Camel naming convention** 85](#_Toc73357557)

[**Recommendation of coding standard** 85](#_Toc73357558)

[**Conclusion** 86](#_Toc73357559)

[**References** 86](#_Toc73357560)

# 

# **Section A**

## **Report**

**REPORT ON ALGORITHM AND**

**ITS APPLICATION**

**DEVELOPMENT PROCESS**

**Anish Devkota**

**BTECH HND**

### **Introduction to Algorithm**

Algorithm is known as a rule or procedure followed for performing calculations and other specific tasks. Algorithm in programming can be defined as the sequence of instruction that helps in solving various problems through step wisely.” An algorithm is the list of instructions and rules that a computer needs to do to complete a task” (ThinkAutomation, n.d.). Algorithm also describes the exact steps to be followed by system to reach the main goal. Simply, algorithms command the computer system how to do the task and make it work according to the command.

“A standard textbook definition would be – an algorithm is a well-defined step by step solution or a series of instructions to solve a problem.”( (Pedamkar, n.d.) Algorithm consists steps like; Problem definition, data collection, data processing, logical approach, solution. Therefore, Programming algorithm is written in simple language and it is not a computer code. Algorithm is a simple human language which has a similarity to pseudocode.

### **Characteristics of Algorithm**

1. Algorithm knows the exact and correct steps to execute.
2. Algorithm ends by giving the expected result after execution of finite instructions
3. Every process is described clearly
4. Algorithm might contain zero or more well defined inputs and gives 1 or more well defined outputs.

Algorithm always give the direction to solution and helps to choose most ideal result. After writing algorithm, flowchart is designed following it and lastly program is build. As, flowchart is the sequential presentation of any program in a diagrammatic form.

### **Advantages:**

* Algorithm can be used for different meaningful task that can be instructed for logical representation.
* Understood able to everyone and written easily.
* Algorithm is also easy to convert into a flowchart and program.
* It is easy to debug as every step has its own logical sequence.

### **Disadvantages**:

* It is time consuming because of steps for producing it into program.
* It is not executable in the programming language
* No syntax to write an algorithm.
* Error detection cannot be easily achieved.

### **Examples of Algorithm with flowchart and its working**

1. Making an algorithm of simple login process;

Step 1: Start

Step 2: Input the username and password

Step 3: Check the username and provided password

Step 4: display invalid for incorrect and move to step 2 but move to step 5 for correct one

Step 5: Display the dashboard after login

Step 6: Stop

Above is the example that represents the simple login process of a system. Now, representing this algorithm to flowchart:

Open dashboard

Invalid

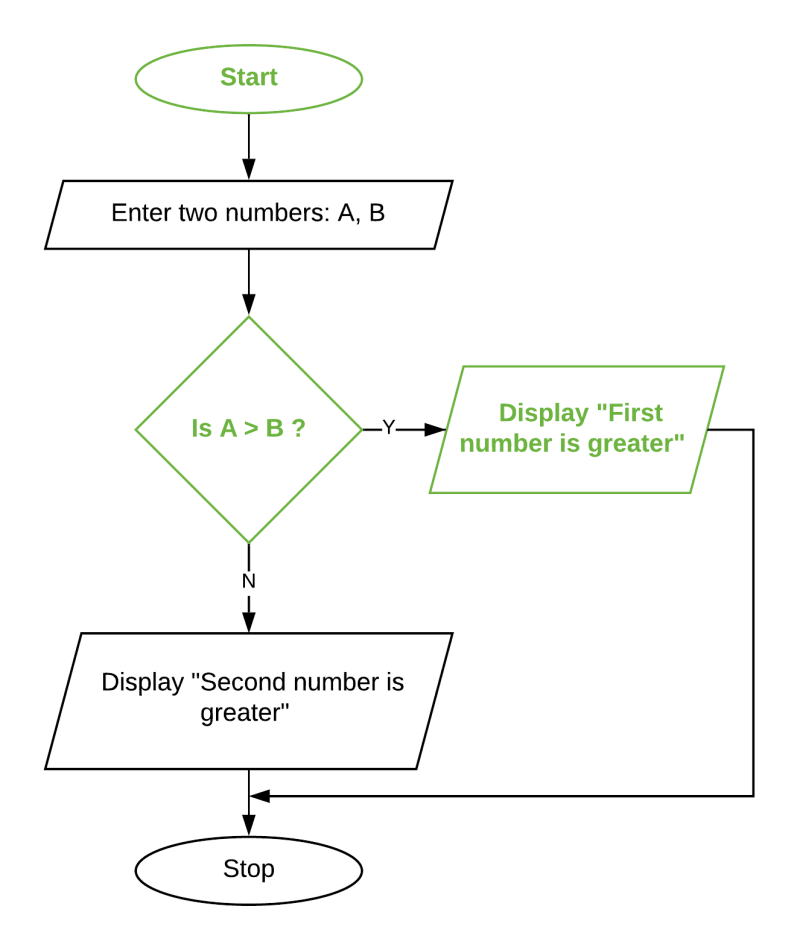
Check name and password

Enter username and password

Working: here, the step will be started and username with password will be given by user. Then, the system checks the data in database. If there is any mistake or invalid data, the process will move again to the input step. And dashboard will open if correct.

1. Algorithm and flowchart to find greater number from two numbers

Flowchart;



Algorithm;

Step 1 = start

Step 2 = enter two numbers

Step 3 = check if a>b

Step 4 = if not true display b is greater and if true display a is greater

Step 5 = end

Working: here, user will input two numbers, then condition of greater will be checked. After being checked, the system will give the number which is greater.

### **Outline process of developing application**

This is a type of process/steps used by the software development offices or industries to make a software application. This also helps in understanding the main goal and cycle of software development. This process mainly defines the activities followed or to be followed for producing the suitable application. Simply, this can be known as the process in building successful application. The major steps in this process are:

* Planning and requirement analysis:

Firstly, to build an application there must be good understanding of the system requirements. The main aim or goal must be well known and the planning must be done according to the possible and effective paths to be followed by the developers. During this planning, the requirements also must be fulfilled. Lastly, defining the requirement and planning for it is done in this process.

* Designing:

After knowing the requirements and plan to implement for building successful application, the system design must be prepared. It clearly defines all the architectural modules of the system. This step/process can also be mentioned as sketching of the planned application.

* Building or developing product

Being able to know the requirements and plans to work for it, design was prepared. Now, the building process of the system will be started The system is developed after evaluating the produced design in a small programs called unit. Then after testing the final product is developed.

* Testing the product:

When the application is built successfully, then there will be the final testing. Meet of requirement, user friendliness, understood designing etc. are checked. After being checked and the application will be deployed in market. Checking the functionality according to the client’s demand is done in this process.

* Deployment in market:

The product will be deployed in a market for the clients/customers. This is also the first stage of the production to the market. The final product after testing and evaluating is deployed in the market.

* Maintenance

After successful deployment, maintenance of the application must be done by the developers. Based on the client’s feedback the application must be improved and maintained up to date. This also helps in development of the application and popular to the market. Adding the new features and capabilities is also done in this process.

### **Development environment**

Development environment is defined as the process or tools used for developing the source code or program involved in making application. Environment where tools and set of rules are coordinated for developers to provide convenient and systematic view of the development process is integrated development environment.

Talking personally, our team have used Visual Studio 19 as my developing environment for successfully building the required application/system. Being easier to use and debug, visual studio was found to be more favorable for us as an application developer.

Some more feature by using this development environment are:

* Easy in debugging the code
* This environment was favorable for editing source code and different scripts.
* This also help in transforming source code into computer language

### **Programming Language**

A set of commands used for creating a software program is called programming language.” A programming language is a computer language that programmers use to develop software programs, scripts or other sets of instructions for computers to execute.” (Computer Hope, n.d.) Therefore, there are various programming languages and all share the similarities of functioning but with own syntax. Some of the popular languages are; C, C++, C#, JavaScript, CSS, Python etc. As, our team were known to the c, C++, C# languages.

For our application building process, we have used C# language. We used this language because it is much easier to use syntax wisely than c and C++. C# is also known to be object oriented with a mixture of procedure oriented. Working on C# is a bit less flexible because it depends on NET framework. Mainly, C# is understood by everyone in our team and it includes simple coding. C# is also a part of .Net framework. C# also produces the efficient programs. Therefore, being more known to the language and the syntax our team conclude of using C# for making the successful application.

Having the best functionality, we choose C# for building our given application.

### **Platform environment**

Computer platform is a computer software or hardware architecture which acts as the foundation of computer system. Environment here is defined as the collective configurations of software, hardware. The .Net framework is a revolutionary platform that helps to write the web applications, windows applications, web services. As, our team have also chosen the C# language for our application development and C# is meant to be the part of .Net framework. So .Net frame work is the platform our team have used for building the given application. This environment also consists of various library of codes used by client language. (tutorialspoint, n.d.). Also having the functionality of ease in accessing by any sort of programming languages makes this platform more reliable.

Lastly, our chosen language (C#) is a part of .Net framework was the platform environment followed by our inventory system development team.

### **Inventory management system design layout**

Our developing team was given the task to prepare an inventory system application for a import export company. The owner requirements were to make application that is able to keep record of products, keep the record of the products on stock based on purchase and sale, stock increased when purchased by owner and decreased when sold.

The starting basic design layout for our application is shown below:

Inventory system

Login

* Purchase
* Sales
* Purchase report
* Sales report
* Category
* Unit
* Product
* Customer
* Supplier

Setup

Transaction

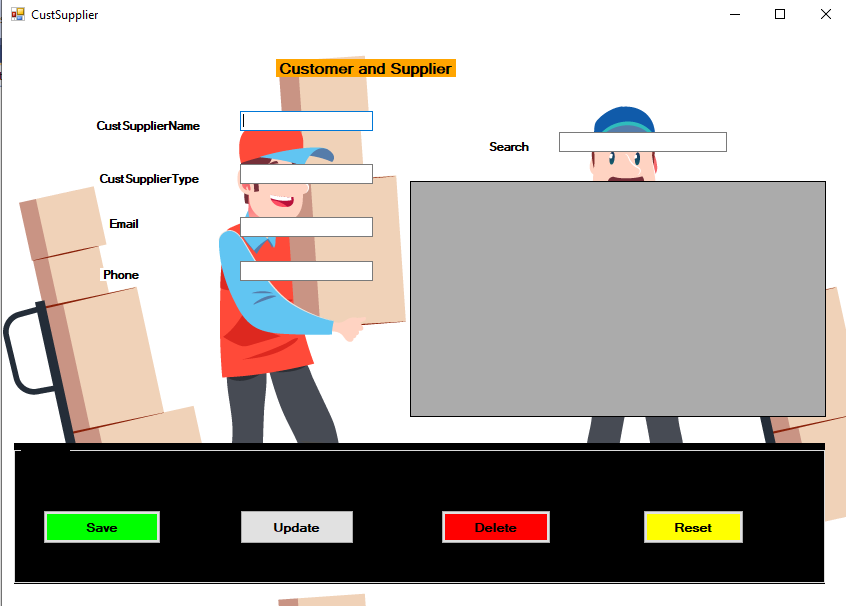
Reports

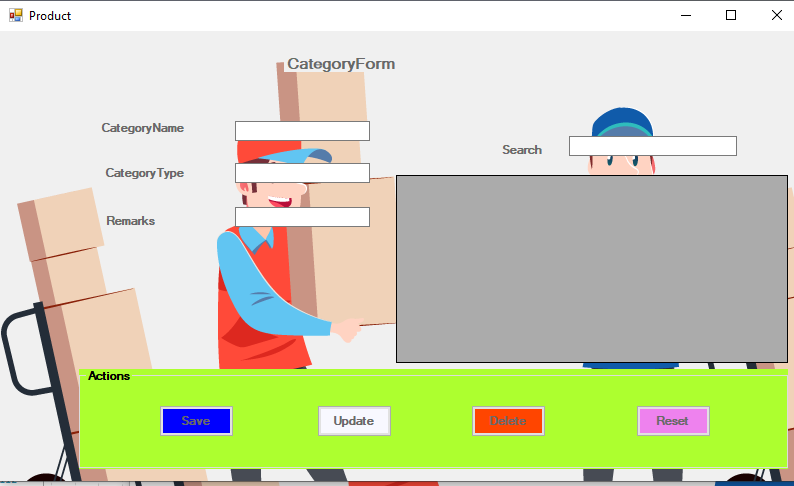
Dashboard

### **Design forms of inventory system created in visual studio 19:**

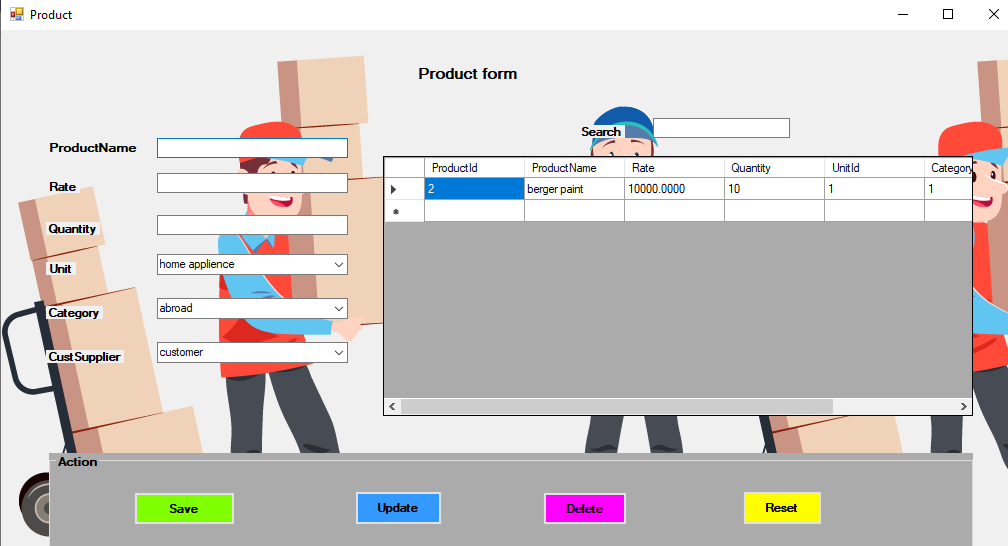


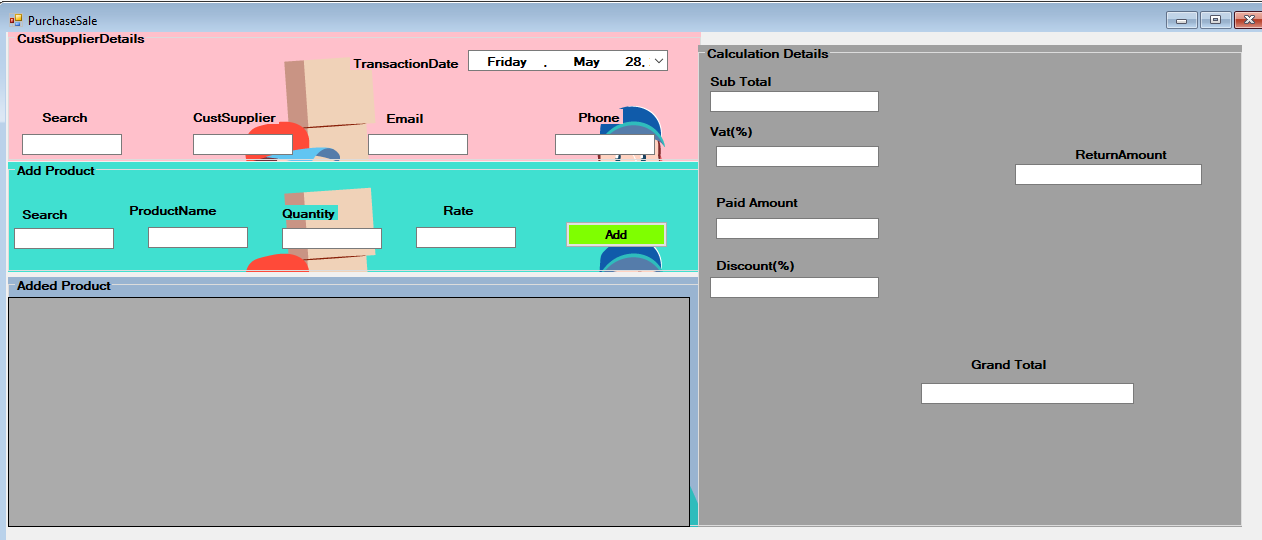












### **Data dictionary:**

Below is the data dictionary of our required application:

User:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Column Name | Data type | Constraint | Remarks |
| 1 | User Id | Int | Primary key | Unique Id |
| 2 | User name | Varchar(50) | Not null | Name of user to login |
| 3 | Email | Varchar(50) | Not null | Email of user |
| 4 | Password | Varchar(50) | Not null | Password of user |
| 5 | Phone | Varchar(50) | Not null | Contact number of user |

Customer Supplier:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Column Name | Data type | Constraint | Remarks |
| 1 | Customer/Supplier Id | Int | Primary key | Unique Id |
| 2 | Customer/Supplier name | Varchar(50) | Not null | Name of Customer/Supplier |
| 3 | Email | Varchar(50) | Not null | Email of Customer/Supplier |
| 4 | Password | Varchar(50) | Not null | Password |
| 5 | Phone | Varchar(50) | Not null | Contact number |

Category:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Column Name | Data type | Constraint | Remarks |
| 1 | Category Id | Int | Primary key | Unique Id |
| 2 | Category name | Varchar(50) | Not null | Name of category |
| 3 | Category type | Varchar(50) | Not null | Type of category |
| 4 | Remarks | Varchar(50) | Not null | Remarks for category |

Unit:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Column Name | Data type | Constraint | Remarks |
| 1 | Unit Id | Int | Primary key | Unique Id |
| 2 | Unit name | Varchar(50) | Not null | Name of Unit |
| 3 | Alias | Varchar(50) | Not null | Also known as name |

Product:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Column Name | Data type | Constraint | Remarks |
| 1 | Product Id | Int | Primary key | Unique Id |
| 2 | Product name | Varchar(50) | Not null | Name of user to login |
| 3 | Rate | money | Not null | Email of user |
| 4 | Quantity | Int | Not null | Password of user |
| 5 | Unit Id | Int | Foreign key | From unit |
| 6 | Category Id | Int | Foreign key | From category |
| 7 | Customer Supplier Id | Int | Foreign key | From Customer Supplier |

Transaction master:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Column Name | Data type | Constraint | Remarks |
| 1 | Transaction Id | Int | Primary key | Unique Id |
| 2 | Customer Supplier Id | Int | Foreign key | From customer Supplier |
| 3 | Transaction date | date | Not null | Date of transaction |
| 4 | Type | Nvarchar(10) | Not null | Type |
| 5 | Grand Total | Decimal(18,0) | Not null | Total transaction amount |
| 6 | Tax | Decimal(18,0) | Not null | Tax in any product |
| 7 | Discount | Decimal(18,0) | Not null | Discount in product |

Transaction detail:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Column Name | Data type | Constraint | Remarks |
| 1 | Transaction detail Id | Int | Primary key | Unique Id |
| 2 | Transaction Id | Int | Foreign key | From transaction master |
| 3 | Product | Varchar(50) | Not null | Name of product |
| 4 | Quantity | Int | Not null | Quantity of product |
| 5 | Rate | Money | Not null | Rate from product |
| 6 | Total | Decimal(18,0) | Not null | Total amount of product |

The above table shows the type of data used in our system for different attributes of a different tables.

### **Algorithm and flowchart of application**

Below is the algorithm and flowchart modules of our designed application.

1. Algorithm and flowchart for the Login button

Step 1: Start

Step 2: Input the username and password

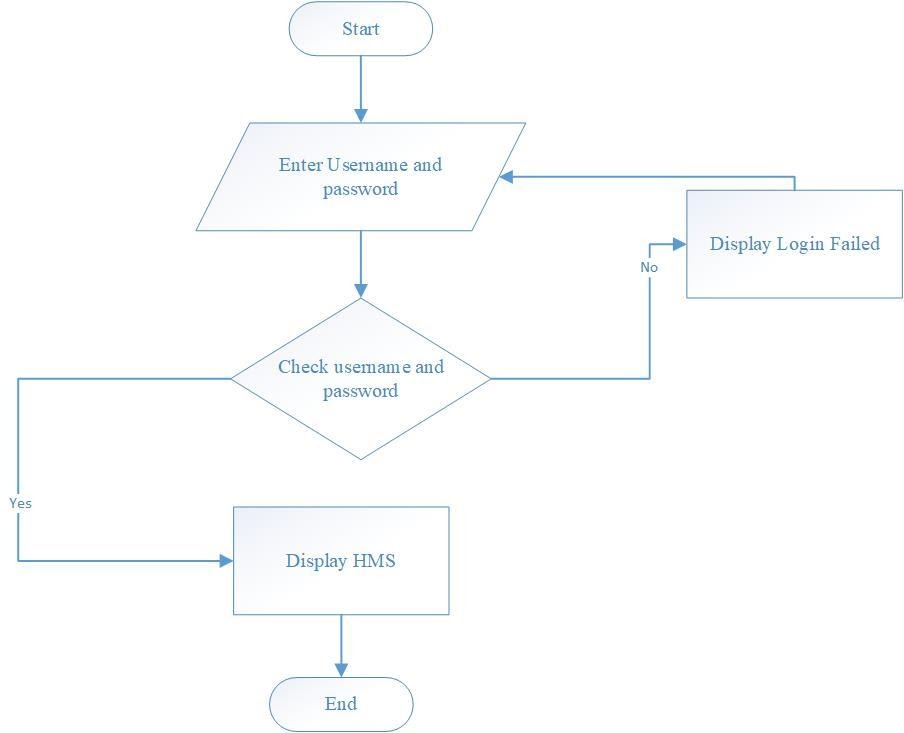
Step 3: Check the username and provided password

Step 4: display invalid for incorrect and move to step 2 but move to step 5 for correct one

Step 5: Display the dashboard after login

Step 6: Stop

Flowchart:

****

1. Algorithm for saving new supplier and customer

Step1: Start

Step2: Enter data for customer and supplier form

Step 3: Click save button

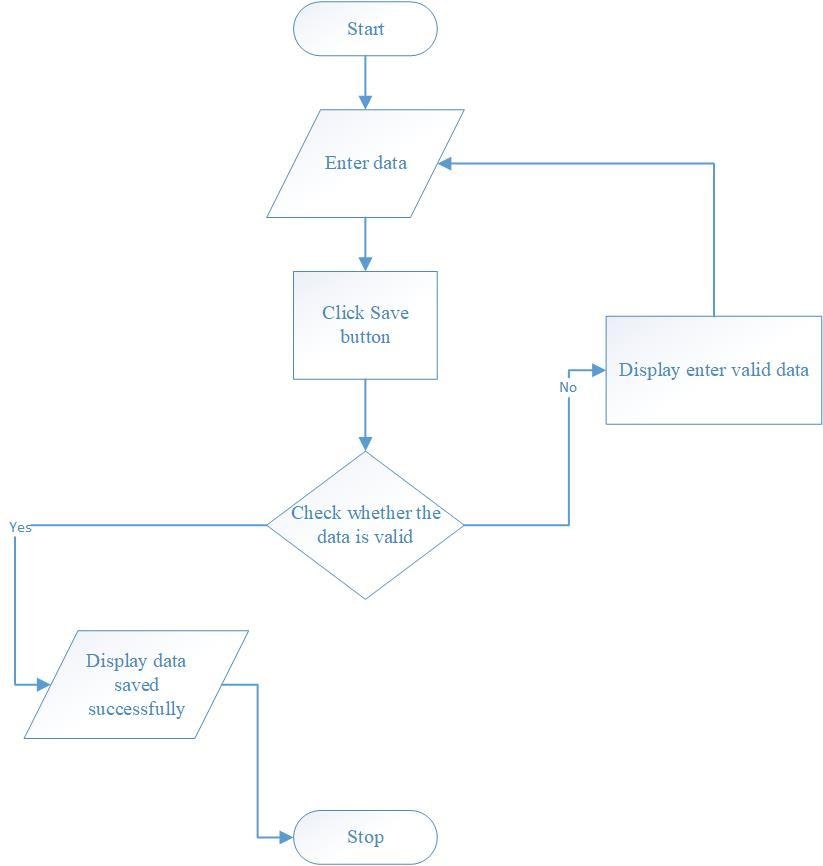
Step 4: Check whether data entered is valid or not

Step 5: Display the data saved successfully for correct data

Step 6: Display invalid for invalid data

Step 7: Stop

Flowchart:



1. Algorithm and flowchart for saving product data in product table

Step 1: Start

Step 2: Enter data of product to store

Step 3: Click save button for saving the data

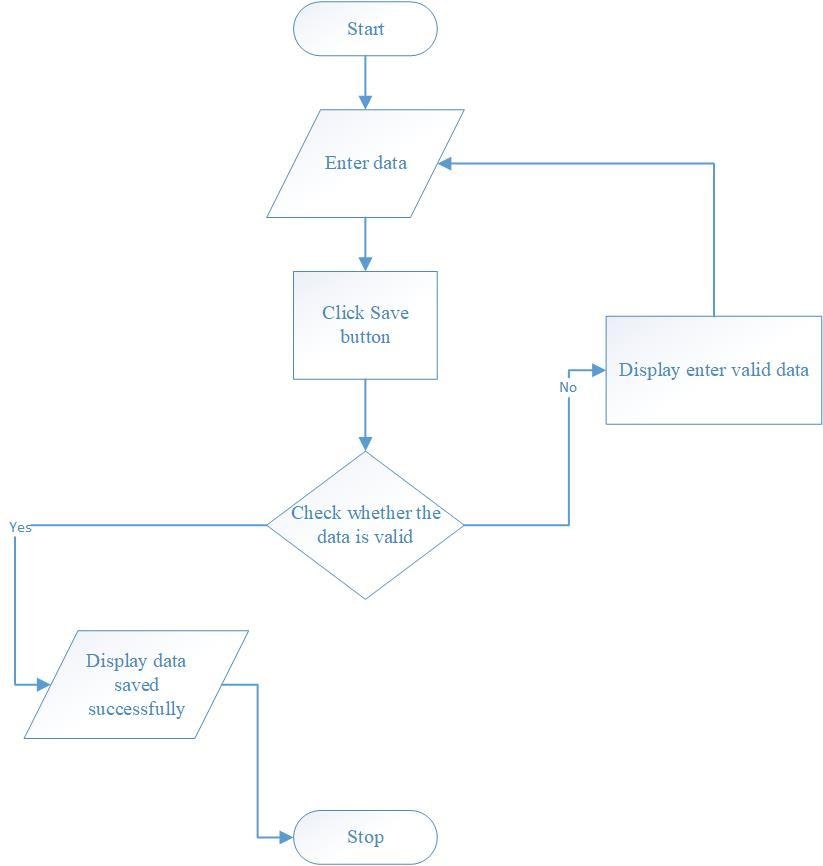
Step 4: Check for the valid data according to the data types

Step 5: Display data saved successfully for valid entry

Step 6: Display invalid for Incorrect entry

Step 7; Stop

Flowchart:



1. Algorithm and flowchart for deleting any components of table using delete button

Step 1: Start

Step 2: Click delete button

Step 3: Check for the confirmation of user

Step 4: If user gives yes then delete and end if no

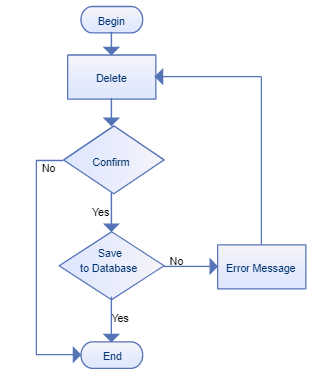
Step 5: Ask for changes in database or not

Step 6: If yes delete the selected component

Step 7: If no is selected then display error message

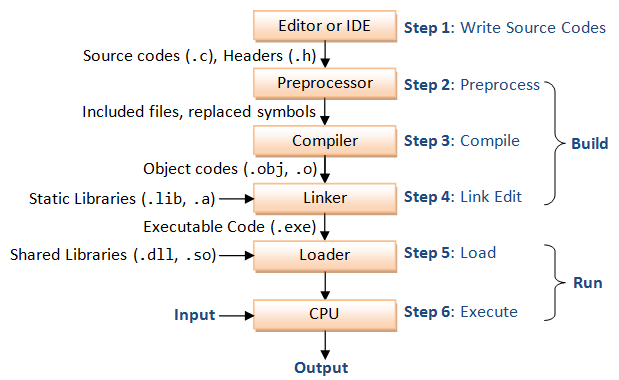
Step 8: stop

Flowchart:



### **Code execution process**

Code execution process where the program is converted to machine language and executed by programmer/ user in understandable way. For completion of any program, language must be understood by machine. So when the language is understood to machine the task is done and executed. And for execution of human language to machine language go through many steps such as Source code, Compiler, Pre-Processor, Linker, Loader, Memory.



The above figure explains the mechanism for code execution process. Firstly, a developer writes the source code and header files. Then the source code is translated by preprocessor as machine level language. For example; #include is an example of pre-processor directives. After being preprocessed, compiler will compile the code into object codes (obj. o) then the linker links the compiled object codes with other library codes and make the code in executable form. Then the loader will load the final executable code into the machine and lastly the output of source code is displayed.

* Source code:

“Source code is the list of human readable instructions that a programmer writes often in a word processing program when he/she is developing a program.” (Bolton, 2019). Source code must be converted into machine level code because this is only readable and understood by human not machine. It is processed through compiler to turn into machine code that is known to computer.

* Preprocessor

The preprocessor directives give instruction to the compiler to preprocess the information before actual compilation starts. All preprocessor directives begin with #, and only white-space characters may appear before a preprocessor directive on a line. Preprocessor directives are not statements, so they do not end with a semicolon (;)” (tutorialspoint, n.d.).

* Assembler/Compiler

The assembler is a process of documenting or converting all the source code into machine level code. Here, the source code is converted into the binary form which is understood to the computer and produce an executable code. Assembler is also referred as a compiler of assembly language which provides services or interpreter.

* Linker

The linker links the compiled object codes with other library codes and make the code in executable form. This is also called as final stage of compilation process. It makes the input from user as one and produce to the loading process.

* Loader

Loader is considered as the final stage of program. Here, the code is executed into the main memory. This is the responsible factor for loading the executable files into the system’s memory. Loader also helps in validate program for memory requirements, permission, initializing the registers etc.

* Memory

This is the final step of execution of any program in computer. Here the program conducted in a machine is saved. The process from source code and loader including its product is saved in memory. In a computer while executing a program, RAM is responsible for the time of producing the final product and ROM is responsible for keeping or storing the final product.

## **Implementation of algorithm**

Hereby after knowing the algorithm of any sort of program we can create a source code for it to apply in program. Algorithm also helps in the code writing and designing process. Understanding of the process of code is also understood by flowchart and algorithm. Below is the implementation of a program with algorithm and flowchart and program written following C language.

Question: Find the simple interest from the number.

Algorithm;

Step 1= start

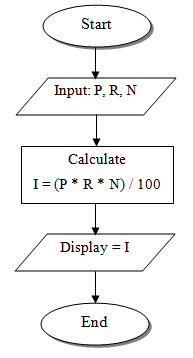
Step 2= get the value of principle, time and rate

Step 3= calculate with the process (P\*T\*R)/100

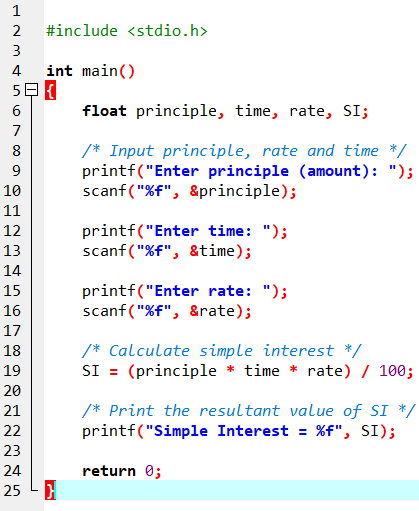
Step 4= display the value of interest

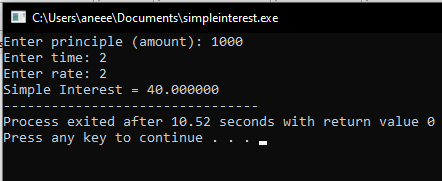
Step 5= end

Flowchart;



Code written in Dev C++ following the C language method:





This is the process of implementation of programming language following the algorithm and flowchart of certain task.

### **Relation between algorithm and source code**:

Algorithm is the first footstep for solving various problems. Algorithm is also the mind makeup for building the successful program. Algorithm also makes the developers easy to learn any sorts of programs. And for our given project, there is a selection of suitable platform, language and development environment. Hence following the components and steps of outline process of application development has been followed by our team.

The code execution process goes through the above mentioned six steps for successfully displaying the final result of code provided by programmer. This also describes the working mechanism of different languages and development process while building the applications/software.

Although for a developer following algorithm is the best way to understand a program and write a source code. It is hard in finding the error and solve the problem. Also algorithm is not executed in programming language too. Programmers also should know the appropriate syntax for developing a program. Following algorithm can also be a little time consuming.

# **Section B**

## **Introduction to Procedural Oriented language**

Procedural language is a type of computer programming language that defines the actions to be followed by computer for solving a problem. This language is also called as a command driven language that works with the state of machine. This language specifies a series of well-structured steps and procedures for composing a program. This language is commonly used in different types of programming languages like; C, C++, Java and Pascal (technopedia, n.d.). This programming language gives user freedom to structure the program solving techniques and again reusing code wherever it is required. Programs are decomposed into the number of sub programs which refers to a function.

## **Features of Procedural Oriented Language**

Some main features of using this language are:

* Most of the function in this language are divided into global data
* Program code is written as the sequence of instructions
* This language function in giving the high overall efficiency.
* This language follow top down approach
* Data is not hidden from the system
* More attention is given for completion of task

## **Limitations of Procedural Oriented Language**

Some major limitations of using Procedural Oriented language are:

* The data in this language imposes large space
* This language is found to be not suitable for applications where time is a critical constraint
* Data in this programming are not secured properly
* Simple error in program may not be encountered and it might run.
* This language is also difficult to relate with real world object

## **Function of Procedural Oriented programming**

### **Basic** I/O

Basic I/O can be defined as the set of rules that helps to input and give output of data which are also known as I/O function. This function helps in getting the values or any characters from user and storing it also by displaying whenever required. This function also functions to read any given input and to display data on screen when there is a need to output the result. Some examples of I/O functions are:

1. Scanf () and Printf (): the printf function helps in printing the required statement or solution as output. And scanf helps in reading its characters itself.

Syntax:

Scanf(statement);

{

Statements;

Printf( statement);

1. Gets () and Puts (): the get function here reads the string value from the user and stores in array. Put is used to print the string and represent the number of characters.

Syntax:

{

Gets(name;)

Statement;

Puts (name);

}

1. Getchar () and Putchar (): The function that is used to gets its own characters through the user and return it to the ASCII is getchar (). The putchar() helps to display characters into standard output procedures.

### **Control Statement**

Control statement is a part of programming language that determines the flow of program order in which instructions must be executed. This sort of statement helps in decision making, performing task repeatedly or jump from one section of code to another (sitepoint, n.d.). This mainly helps in choosing a direction of program according to given parameters or conditions. Some types of control statements are:

* If else statement:

This statement is used for finding out the exact solution after solving of problem.

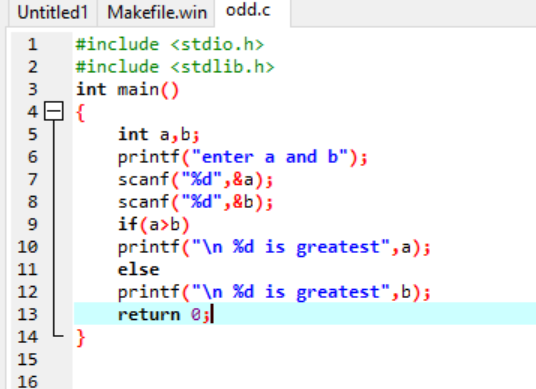
Syntax:

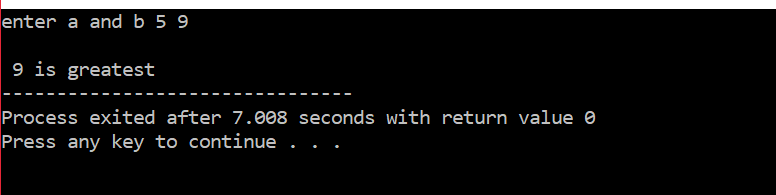
If (condition)

{Statements;}

Else

{Statements}

****

****

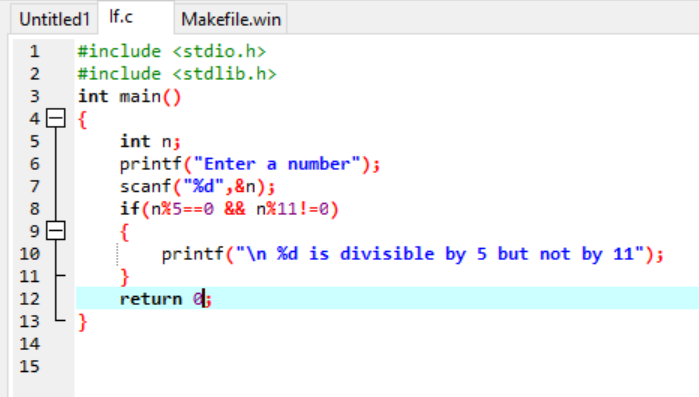
* If statement:

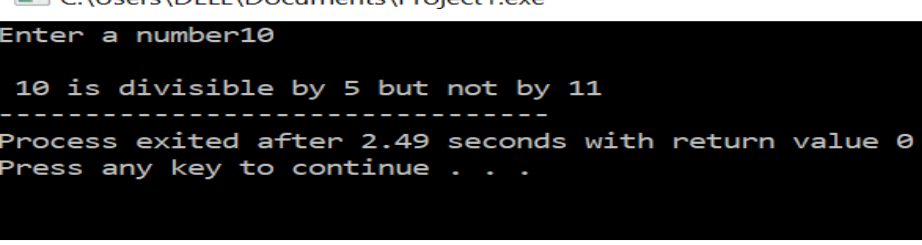
This statement is used for displaying only the solution which meet the given condition of program.

Syntax:

If (condition)

{Statement}





* If else if statement;

This statement helps to check with different conditions and display the exact solution which meets the condition

Syntax:

If (condition)

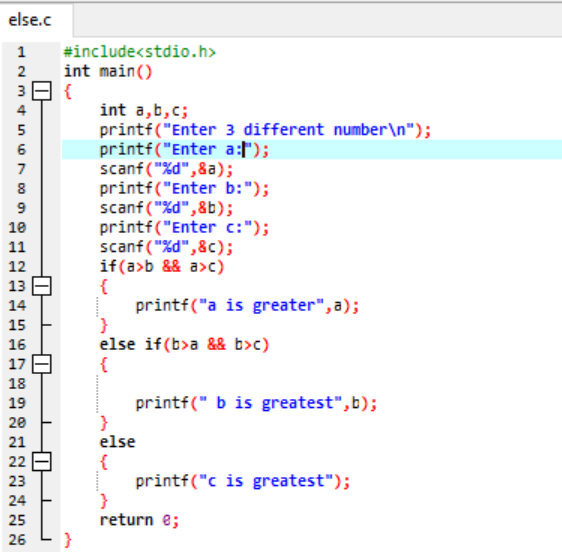
{Statement}

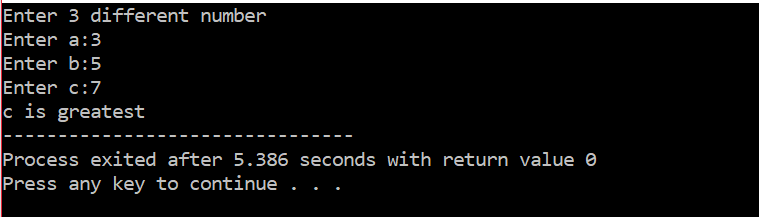
Else if (condition2)

{Statement}

Else

{Statement}



****

* Nested if else statement:

This statements helps to check the different condition and again the if else condition is checked inside the main if else condition.

Syntax:

If(condition)

{If(condition)

{statement}

Else

{statement}}

Else

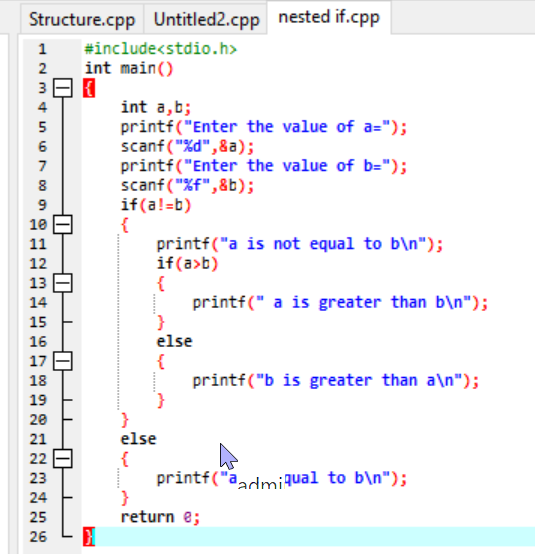
{if(condition)

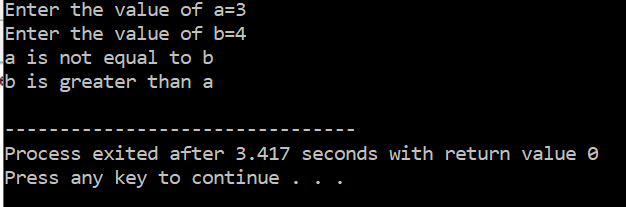
{statement}

Else

{statement}

}



****

### **Looping**

Looping in another word also known as control structure functions in performing the repeated task for the user demanded times. This statement helps to decrease in written syntax by helping to display sequence of statements many times until the condition is false

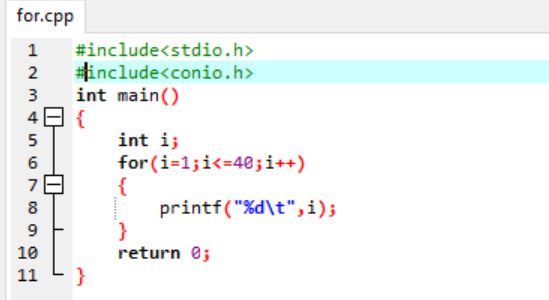
Some looping statements are:

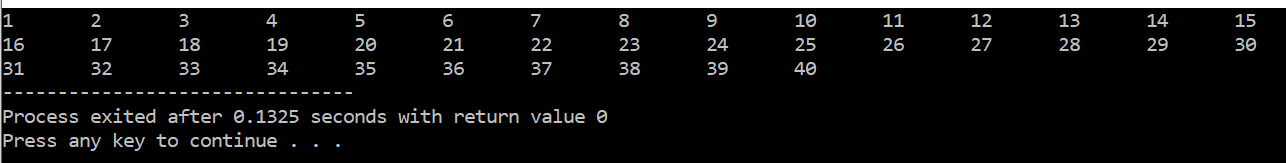
* For loop: this looping statement is mainly used for executing statement before a block of statement for securing number of periods.

Syntax:

For (initialization; condition; increment/decrement)

{Statements;}



****

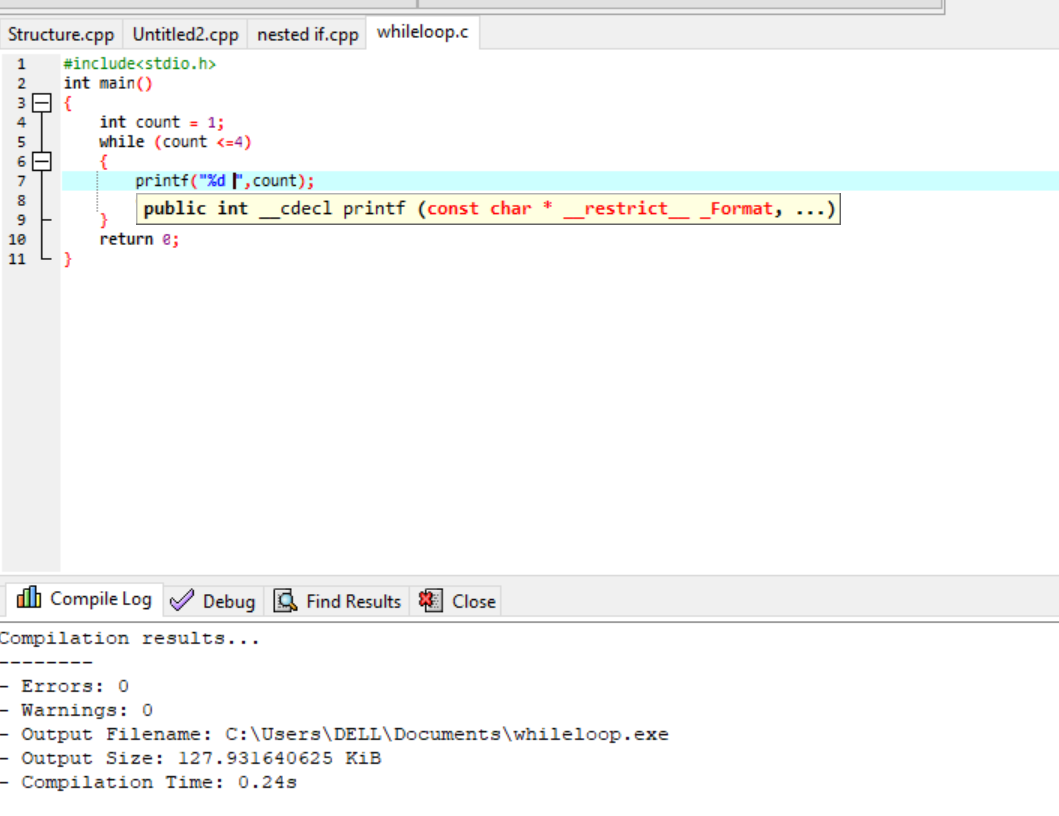
* While loop: this loop is specially works by checking the condition first and implementing the task accordingly. If condition is wrong, then operation is not performed.

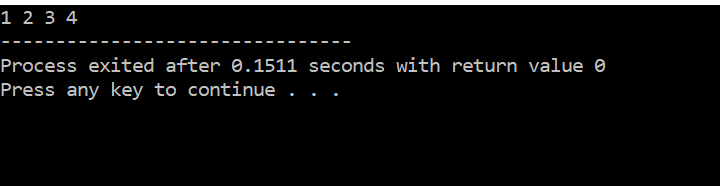
Syntax:

While(condition)

{Statement;

Increment/decrement;}



****

* Do while loop: This loop helps in operating the command for first and after increment/decrement, condition is checked and implemented according to the given condition

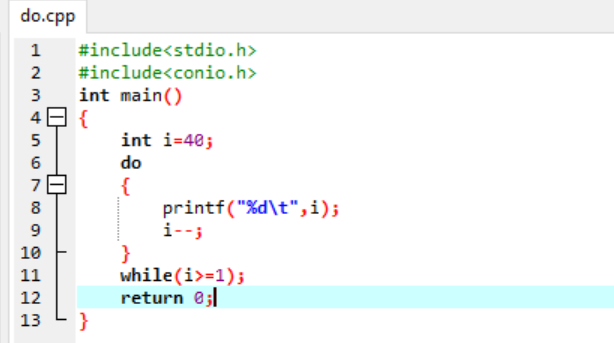
Syntax:

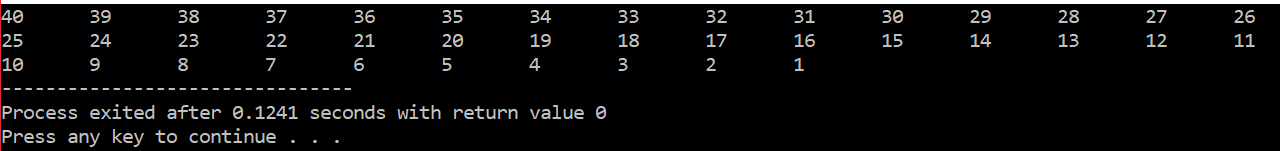
Do

{Statement;

Increment/decrement;}

While(condition)

****

****

### **Array**

In Programming Language an Array is sequence of objects where all of them are same type and size. For Example: - An array of characters, an array of integers or an array of anything that has referred to data type. Array is also known as variable which can hold similar multiple value under same name. Example: a [10] =0,1,2,3,4,5,6,7,8,9.

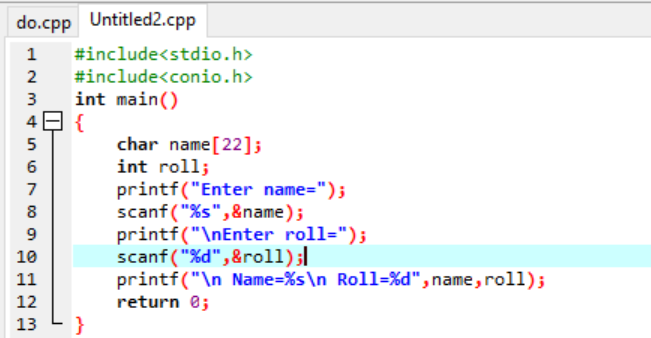
Types of array are:

* One dimensional array

One dimensional array is a container of elements with a single row and stored one by one.

Syntax

1. Int num [100];
2. Float temp [20];
3. Char ch [20];



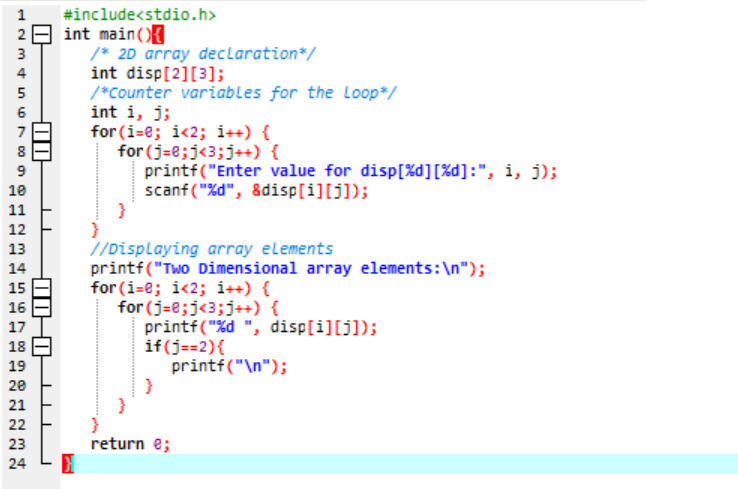
* Multi-dimensional array

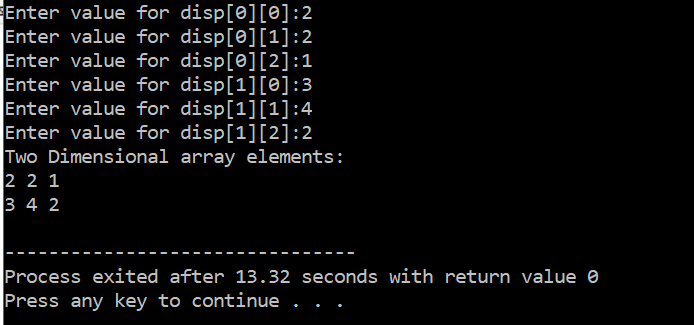
This is a type of array which stores the elements in a tabular or matrix form with row and column but gives the single row value after calculation. Example; a [2][1]

Syntax:

Int da [2][3] = {{1,2,3}, {4,5,6}};

i.e. Data \_ type array\_ name [ row size] [column size];



****

### **Structure**

Structure is a type of user defined data that helps to connect the data item with various nature. Structure consists a single name with numbers of compiled files.

Syntax:

Struct\_name

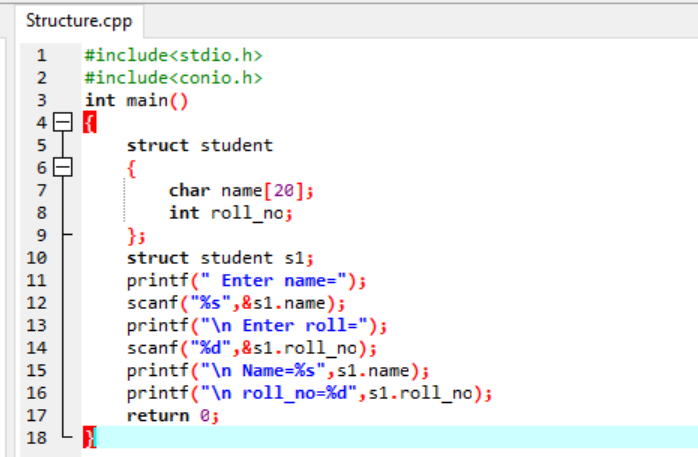
{member variable 1

Member variable 2

Member variable 3

…..}

Struct\_variables



### **Function**

Block of organized code that is used to perform single related programs for multiple time by simply calling it and without writing the code time and again is called function in programming language. There are two types of function:

1. Not returning function:

This is a type of function that does not return any sort of value. This type of function is specially used to display the data without returning value. For displaying the data void is used.

Syntax:

Void PrintName ()

{

Statement;}

Int main ()

{call function;

}

Output: displays the called function

1. Returning function:

This function is used for the returning type or solving type of programming. This function process the condition of function and whenever called it gives the result.

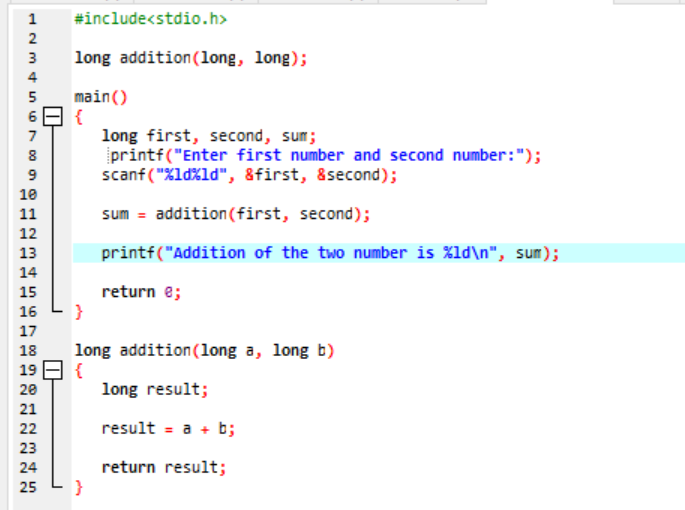
Syntax:

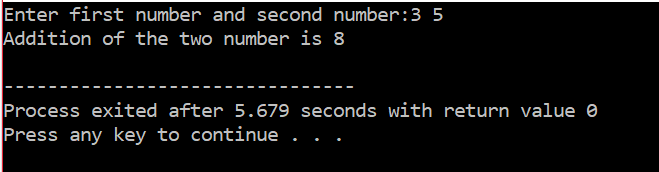
Return type function name (list of arguments)

{//statement

Return value;

}





### **Pointer**

Pointer is a type of programming language object that helps to store memory address of another value that is located in memory of computer. Pointer are special variables used in storing memory address rather than values. The main purpose of pointer is to save the memory space and gain execution faster. “A pointer can also be used to refer to another pointer function. A pointer can be incremented/decremented, i.e., to point to the next/ previous memory location” (guru99, n.d.).

### **String**

String is a sequential character which is concluded with null character. Strings in other hand is also defined as an array of characters in program. String is represented by using double quote marks.

Example; str = “Hello World”

Syntax: Char string\_name[string\_size]

### **File** **handling**

File is a documentation of related data in the form of bytes on disk. Simply file handling in procedural oriented language stands for creating files, opening the needed files, reading and writing a file, moving to location in file and closing of a file. After execution of program, it will be saved and documented as file. The opening, modification or editing process are done by the help of file handling in programming.

## **Introduction to Object Oriented Language**

“Object Oriented Programming Language (OOPL) is a programming paradigm that relies on the concept of **classes** and **objects**. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects. There are many object-oriented programming languages including JavaScript, [C++](https://www.educative.io/blog/how-to-learn-cpp-the-guide-for-beginners), [Java](https://www.educative.io/blog/object-oriented-programming-concepts-java), and [Python](https://www.educative.io/blog/how-to-use-oop-in-python)” (Doherty, 2020). This language was established to overcome the constraints of procedural oriented language.

OOPL consists simple structures and objects of OOPL can be used across programs. This language is also found to easier to debug and can be secured by the help of encapsulation process.

## **Feature of OOPL**

Some of the main features of OOPL are described below:

### **Classes**

Classes are meant to be known as a blueprint of any object in programming. Classes are also described as an user defined data types. It consists the detailed information or structure of a real object program. Talking practically, class can also be called as an architecture or map for creating house. Likely, this is a component or map of real program to be done.

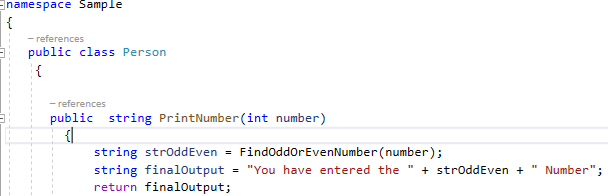
Syntax:

Access modifier class ClassName

{

//statement or data

}



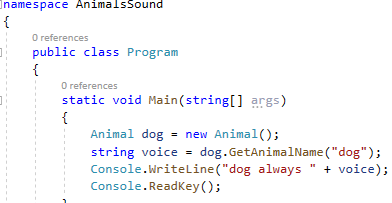
Here the class name of program is person. In this class, the function of object is mentioned. For displaying the output the characters or functionality of class is called in an object and displayed.

### **Object**

In OOLP, object is defined as the occurrence of the structured class. As per real world scenario, object is just like the house and class a blueprint of it where house is structured or build according to the blueprint. Exactly the same, class and object works together for building a program. Object are also known to be the real existing block of program.

Syntax:

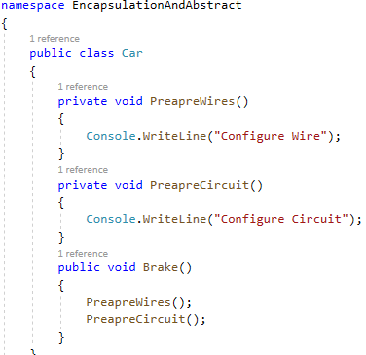
ClassName ObjectName = new ClassName();



### **Data encapsulation**

Simply, encapsulation is the process of containing an information or data in object and only displaying the selected information in OOPL. While establishing a program, there might be some information that is not important for the users to display. So, for hiding the process which clients does not require can be encapsulated with in a single class using a private access modifier inside of public modifier. The encapsulated data are not visible to the clients or users but clear to the developers.

Syntax:



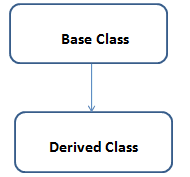
Syntax and example of data encapsulation.

### **Inheritance**

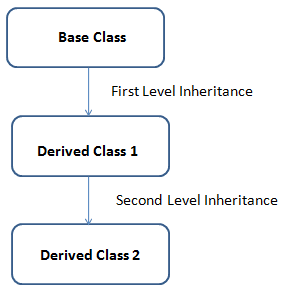
Scientifically inheritance means to contain or inherit the behavior or parents or old generation. Likewise, in OOLP inheritance is the process where child class inherits the behavior or functions of the parent class and make their own functionality. This process is mainly used when there is a case of reusability. Applying the process of inheritance, child classes automatically can access to the behavior of parent class.

The types of inheritance process in OOLP are mentioned below.

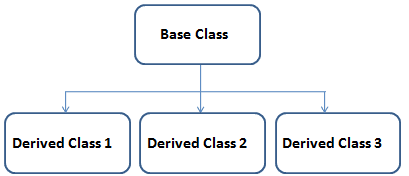
Single level inheritance is one of the type of inheritance where succeeding between only one child and base class is held.



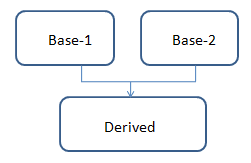
On the other hand, Multilevel inheritance can be access in succeeding from one derived class to another derived class.



Hierarchical inheritance is another form of inheritance which helps different child classes to abstract the behavior of main class.



Multiple inheritance is a inheritance that allows developer using OOLP to build a final class from combining different characteristics of different base classes.



### **Polymorphism**

Polymorphism is meant to be designing objects to share different behaviors. This is the process where many methods can do the same task. Also this process is meant by creating or developing many programs from single object. The two types of polymorphism process are described below.

#### **Static polymorphism**

It is a type of polymorphism which follows the method overloading process. Static polymorphism is also called by early binding step or compile time process. This method contains same name but a different parameter, signature etc.

class DemoOverload{

public int add (int x, int y){ //method 1

return x+y;

}

public int add (int x, int y, int z) { //method 2

return x+y+z;

}

public int add(double x, int y) { //method 3

return (int)x+y;

}

public int add (int x, double y) { //method 4

return x+(int)y;

}

}

class Test{

public static void main (String [] args){

DemoOverload demo=new DemoOverload ();

System.out.println(demo.add (2,3)); //method 1 called

System.out.println(demo.add (2,3,4)); //method 2 called

System.out.println(demo.add (2,3.4)); //method 4 called

System.out.println(demo.add (2.5,3)); //method 3 called

}

}

#### **Dynamic polymorphism**

Dynamic polymorphism also known as runtime polymorphism follows the method overriding process. A child class using this method can provide different functionality than its parent class.

class Vehicle{

public void move(){

System.out.println(“Vehicles can move!!”);

}

}

class MotorBike extends Vehicle{

public void move(){

System.out.println(“MotorBike can move and accelerate too!!”);

}

}

class Test{

public static void main(String[] args){

Vehicle vh=new MotorBike();

vh. move(); // prints MotorBike can move and accelerate too!!

vh=new Vehicle();

vh. move(); // prints Vehicles can move!!

}

}

## **Event driven programming language**

A designed program for responding to user engagement in different forms is event driven programming language. Event driven programming is also called as programming paradigms. The flow of program in this language is decided by the event which are monitored by an event listener named code. Event-driven programming serves the user with the quickest and most accurate responses, which typically translates into better user experience for any given app or program ( (edgetechacademy, n.d.) event driven program is also used and demanded widely in modern programming sector.

## **Characteristics/Features of event driven program**

Some of the important features of event programming are given below:

1. This programming language is found to be more service oriented. This programming helps in providing the each and every services of screen to the user. The task will be executed according to the user demand or request.
2. One paradigm of event driven programming is time driven where code runs accordingly to the triggered time. For example if an program is developed and functioning with checking update for daily, weekly or according to user launch and finally work according to user need or command. This is the time driven paradigms of event programming.
3. Event handlers is also a method of event driven programming which operate an action when certain event is done. , for example; user will execute the code and stop it with clicking the button from the screen.
4. Trigger function is also one of the features of event driven programming that helps in making decision for the specific event selected from event handlers.
5. Events here includes mouse, computer keyboard and user interface which are required for what to trigger in a program. Triggering the events helps in executing process or interacting object in program.

## **Advantages of Event driven programming language**

* One main advantage of using event driven programming is its flexibility. This paradigm also helps developers in altering or applying any changes to the code or content easily.
* This programming language is found to be suitable for graphical interfaces. This program helps in allowing user to select different tools or buttons, put object and edit in own direction.
* This language is found to be more compostable and understood able to user. Also it is easy to find the dividing lines.
* This language also functions in making a programming easier because of direct edit and the predictive coding.

## **Disadvantages of event driven programming language**

* Here the main disadvantage of this programming is to be less logical and obvious.
* Program with the complex GUI might get slower while loading.
* Debugging or error finding is hard in this programming language
* Program including too many forms will be more confusing for the users.

## **Relationship between three programming language**

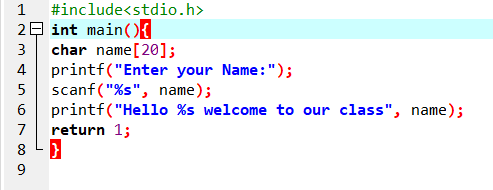
|  |  |  |
| --- | --- | --- |
| **Procedural programming language** | **Object oriented language** | **Event driven programming language** |
| It focuses on sequential execution of process | It focuses on object or data and facilitate to secure it from unauthorized access | It focuses on selecting user interface |
| It provides character user interface to write commands | It provides command writing in modules | It provides GUI to create program |
| It is difficult create any data type | It is not suitable for the small program | It would be failure in certain case |
| Example; FORTAN, C, COBOL | Example; C++, PHP, Java and Ruby | Example; Visual basic, C# |

## **Critical evaluation of source code of application**

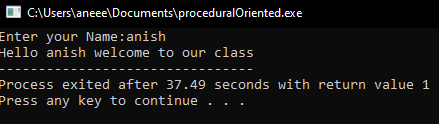
For critically evaluating the source code of application we created a scenario to find the simple interest from the given number. We intended to do this for finding the positive and negative aspects of source code application for different programming paradigms.

Solving the problem with procedural oriented programming:

**Source code**

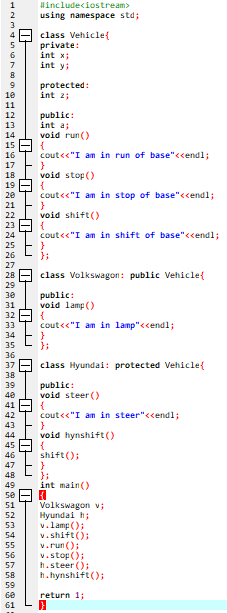
****

**Output**

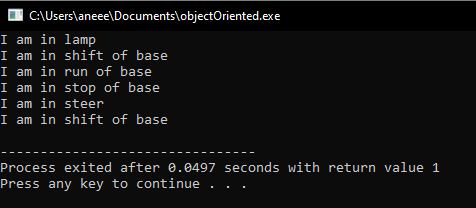


Solving the problem with object oriented programming:

**Source code**



**Output**



Solved by event driven programming:

**Source code**

use Libraries.Curriculum.TurtleProgram.TurtleGame

use Libraries.Interface.Events.KeyboardEvent

use Libraries.Interface.Events.KeyboardEvent

class Main is TurtleGame

action Main

StartGame()

end

action TurtleCommands

SetStartingCoordinates(5,4)        //setting the starting position

//of the turtle

end

action PressedKey(KeyboardEvent event)

if event:keyCode = event:UP

MoveForward()

elseif event:keyCode = event:LEFT

elseif event:keyCode = event:RIGHT

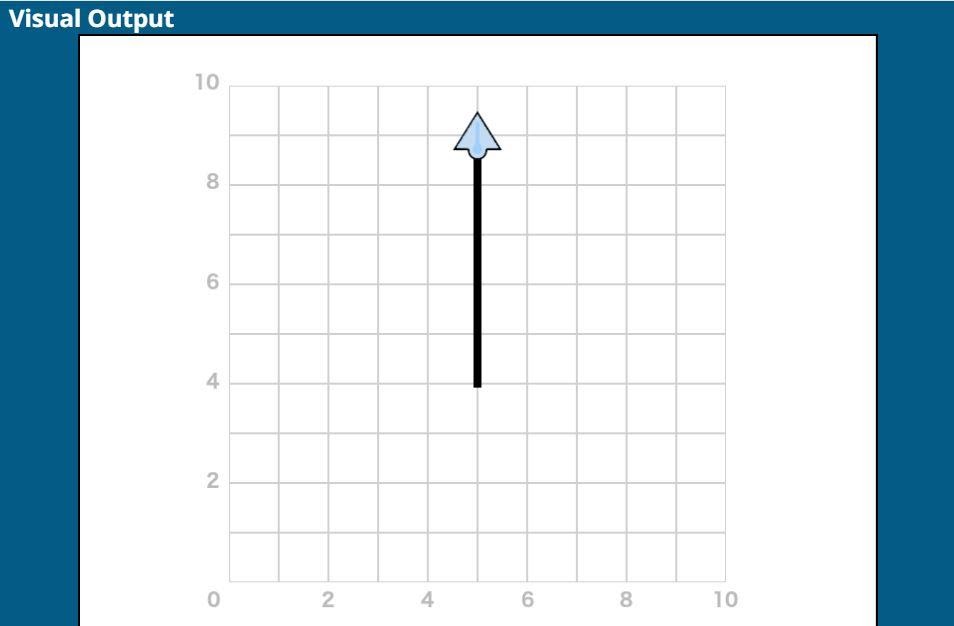
elseif event:keyCode = event:DOWN

end

end

end

**Output**



## **Justification**

Form this problem, we found out different working functionality of different programming paradigms. All of them provide the different languages but acted the same for giving the result. Choosing the well-known paradigms will lead in successful programming. For our team, being more familiar to the object oriented and procedure oriented we analyze these languages to be easier.

## **Introduction to IDE**

Integrated development Environment (IDE) is an application that manages the editing, compiling, running and debugging programs. In simple word IDE is an application that helps in development of software or application. By combining the development tools to GUI this application functions in software development. IDE is determined as the different tool which has a graphical interface workbench and helps to analysis the application

## **Benefits of IDE**

“The overall goal and main benefit of an integrated development environment is improved developer productivity. IDEs boost productivity by reducing setup time, increasing the speed of development tasks, keeping developers up to date and standardizing the development process” (veracode, n.d.).

Some of the benefits of IDE are mentioned bellow:

* Setup will be faster and time saving.
* Helps in project management and the resources are easily available
* Can write different programs with the available tools
* It follows the continual learning process or being up to date with new features and functions.
* It helps in maintaining smooth development cycle

## **Tools available in IDE**

Tools featured by IDE are given below:

* Code editor:

This is the tool of IDE where the main source code is written. This tool is designed for writing and editing the source code. Auto complete is a type of code editor that helps to reduce the time of writing code by suggesting or predicting text. Secondly bracket matching is a editor that helps to determine the enclosing of certain code this also helps in finding the missing bracket and block of code. Syntax check also act as editor by recognizing the correct syntax and highlight the error in program.

* Compiling:

Compiling is a tool used by IDE applications mainly for documenting the source code and translate to computer in machine language.

* Debugging:

This tool in IDE helps to recognize the error in a source code and provide suggestion to required error.

* Documentation:

Documentation tools in IDE help to generate a documented data extracted from the database. This tool also helps to describe tables and columns and database objects.

* Libraries:

These tool provides functionality of providing the functions not included in a core programming language

* Automation tool:

Automatic tools are programs that automate the creation of executable applications from source code.

**.Net database connectivity (ADODB)**

Connecting a program or application to the database is the most needed thing to do for completion of successful application. Talking about the project we also have worked by following .Net framework and database connection was also required for our application. For storing data, updating, editing or deleting we needed to connect the database to our application program. Also being the requirement of Inventory system we established the connection of database and .Net framework.

**Steps for connectivity between .Net and database.**

**Step1: Preparing connection string**

String connectionstr =” Data Source=’Server Name’; Initial Catalog= ‘Database Name’; Integrated Security=true;”;

**Step2: Preparing SQL Connection**

Using (SqlConnection con = new sqlConnection(connectionstr);

**Step3: Open Connection**

Con.Open();

Step4: Prepare SQL Command

SqlCommand cmd = new SqlCommand ();

Cmd.connection= con;

Cmd.CommandType = CommandType.Text;

Cmd.CommandText = @” Your Command Query for Database”;

Cmd.Parametres.AddWithValue(“@Value1”, value1.Text)

Cmd.Parametres.AddWithValue(“@Value2”, value1.Text)

Cmd.ExecuteNonQuery();

After following the above steps connection between Database and C# is completed. There are other steps to fetch data from the database like fetching data which is mentioned in the steps below:

**Fetching Data from Database:**

**First 3 steps are same as above.**

**Step4: Prepare SQL Command**

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable ();

Da. Fill(dt);

## **Conclusion:**

After using the above mentioned functionality and characteristics including types of programming language, we were able of completing the assigned task to develop the application “Inventory Management System”. Basic concept of IDE with the tools and features was also been described above. This concept provided us the knowledge of using the features of IDE application for completion of our project. The implementation of program for our software was then build with this supportive knowledge.

# **Implementation of Application**

After the study or information of different programming paradigms, languages, platform and function of algorithm with development environment, our team were able to implement it for our application development. Our project was to build Inventory Management System. For building our application we choose C# programming language with the help of Visual Studio (IDE application). Then we also use MySQL for storing the required data for the application. And also we need the database for storing data and excessing the file after execution of program.

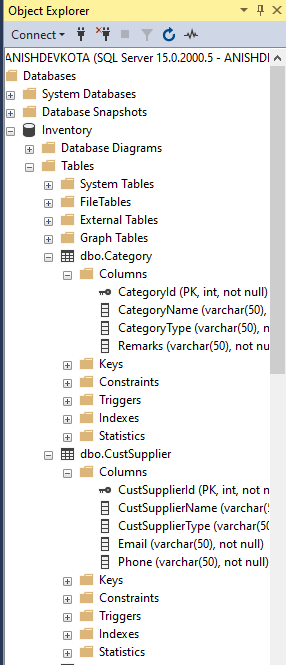
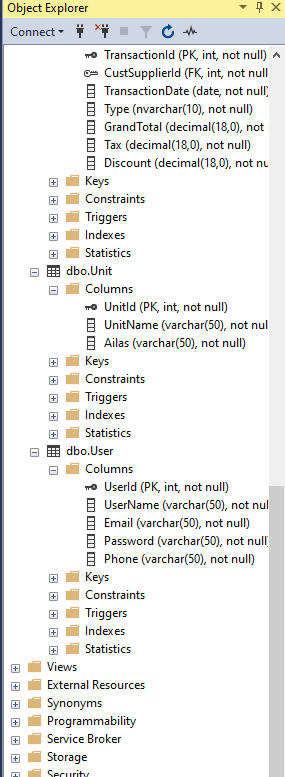
 In this task we will be explaining more and in detail about the debugging process and its facilities in IDE that we had used to make our Inventory System project. Here is an explanation about the concept of coding standard that we had used in our whole application. And we will be evaluating the debugging process for our Inventory Management System we have developed.

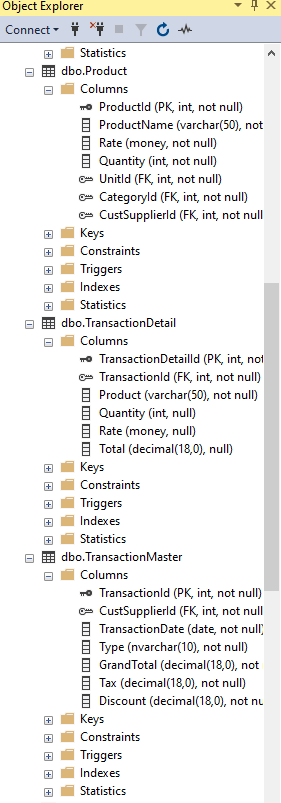
# **Design and code used for inventory system**

We were able to implement our knowledge in our Inventory System. Below are the design and code used for execution of our application as screenshot.

Below is the data table with data type of our inventory system:

The screenshot shown below are the table for forms with its components implemented in MySQL database software for data base connection in our source code. Table shown below also represents the foreign key and primary key of table.



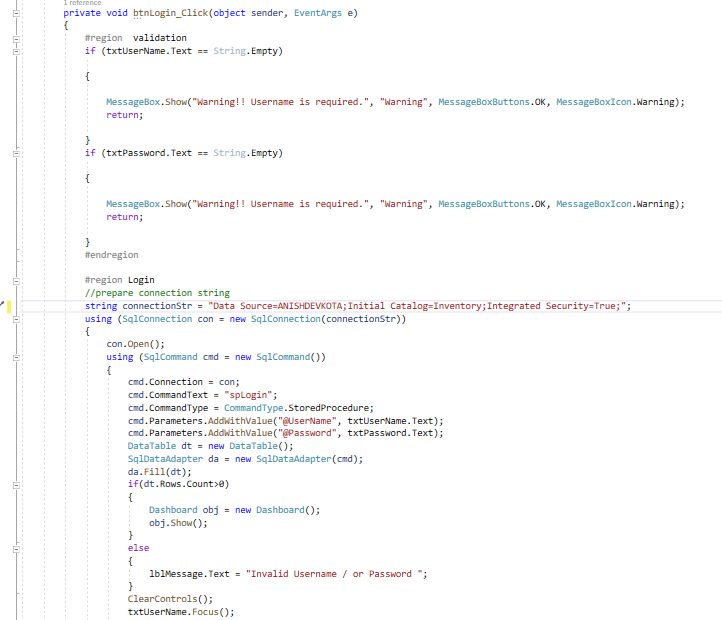
Login with encrypted password:

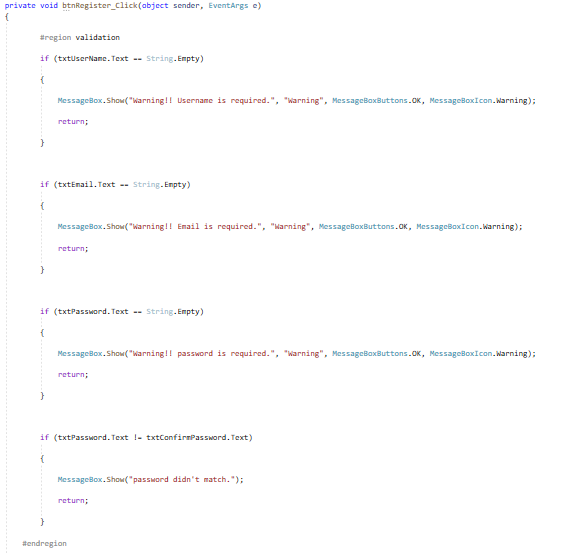


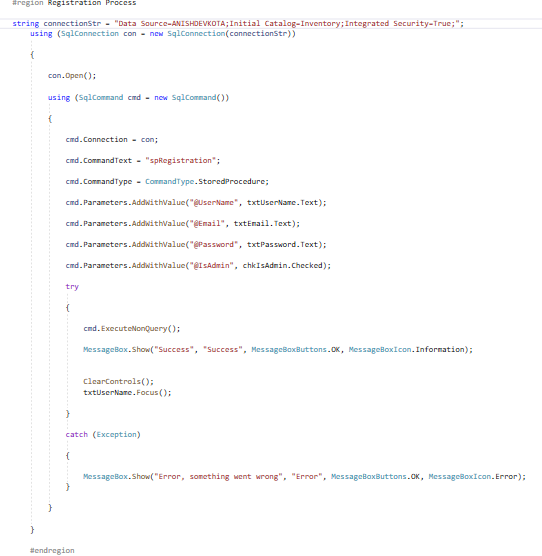
Registration process:



Code:



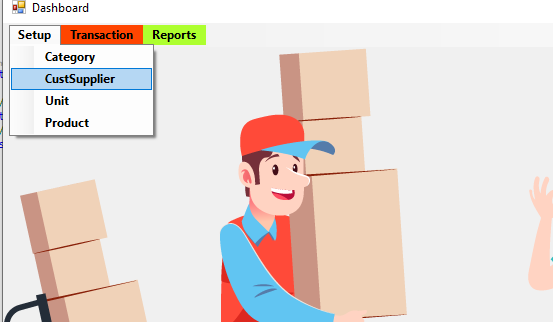




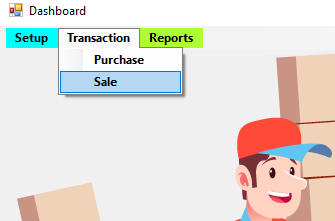
Dashboard:



Setup Menu:



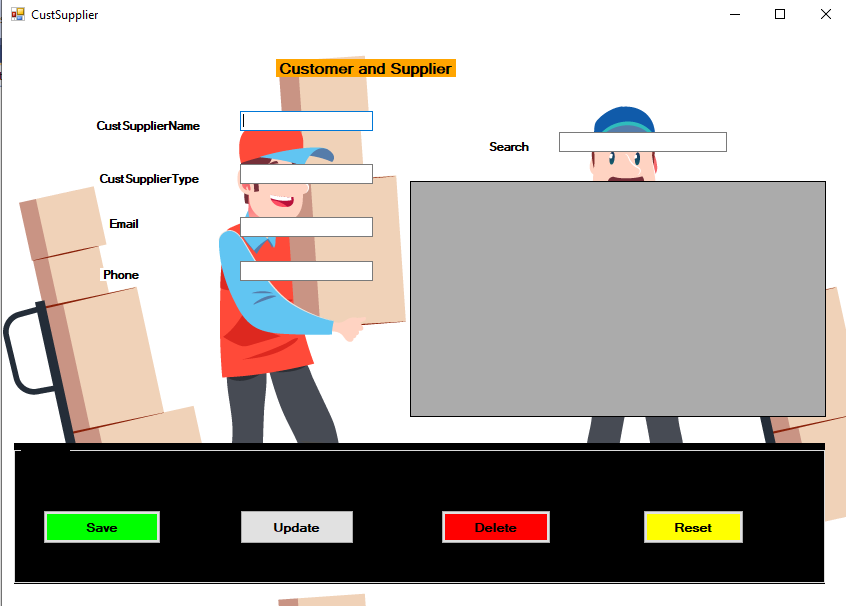
Transaction menu:



Code of dashboard:



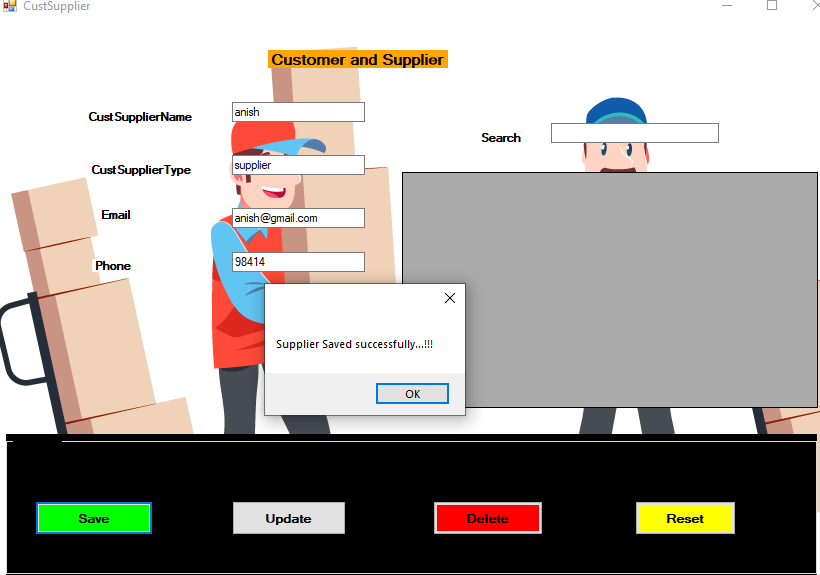
Customer Supplier Form:



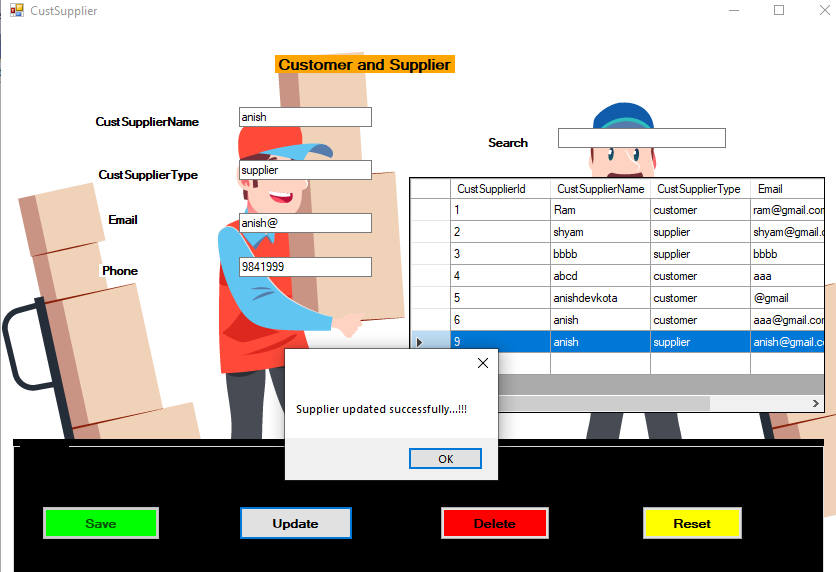
Save, Delete, Update, Reset Operation:

Result with all buttons operations:

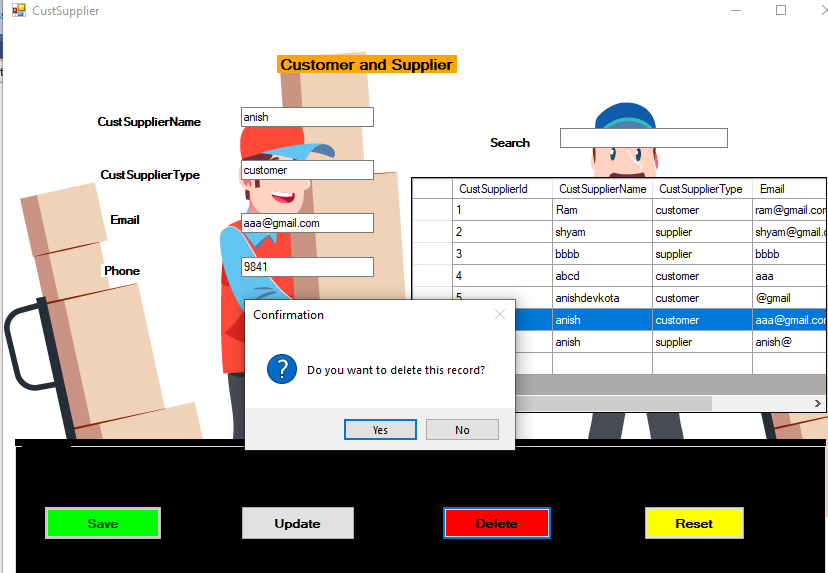
Save:



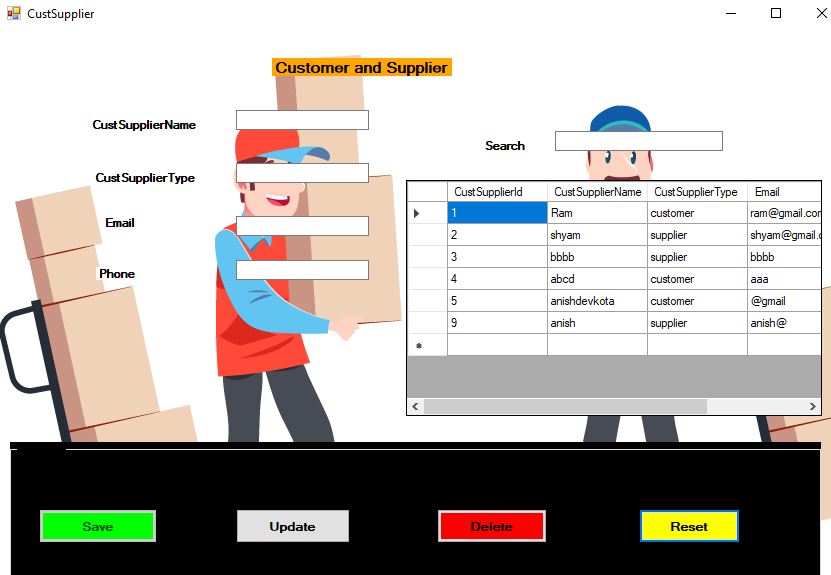
Update:



Delete:

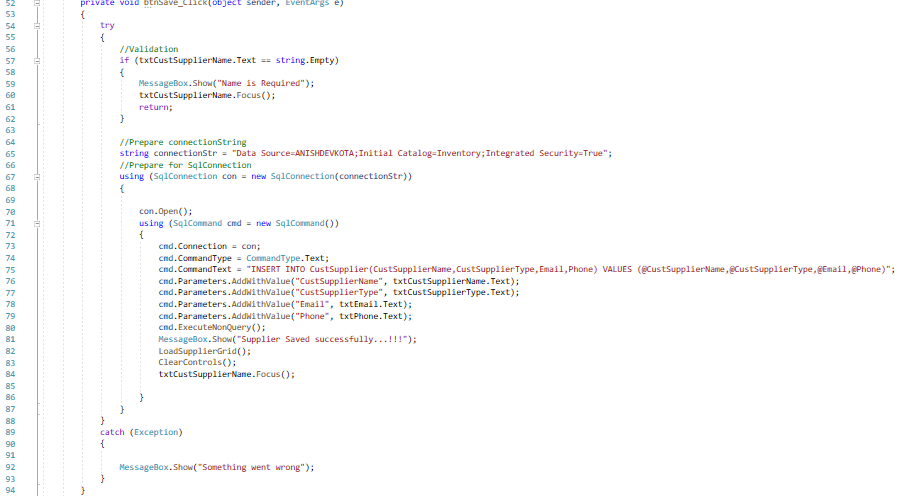


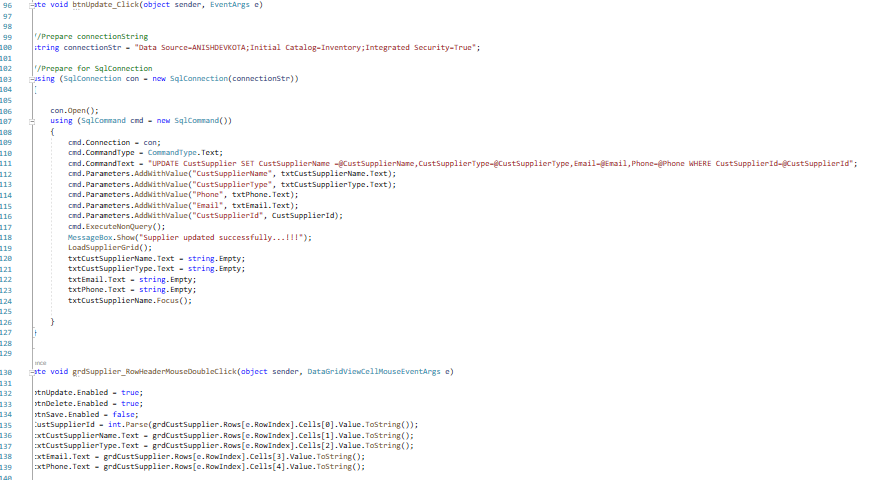
Reset:

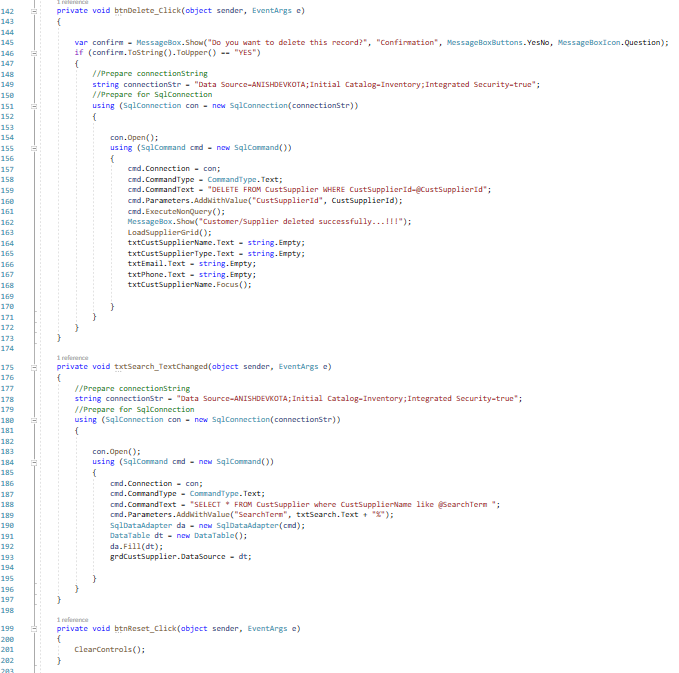


Code of Customer Supplier:

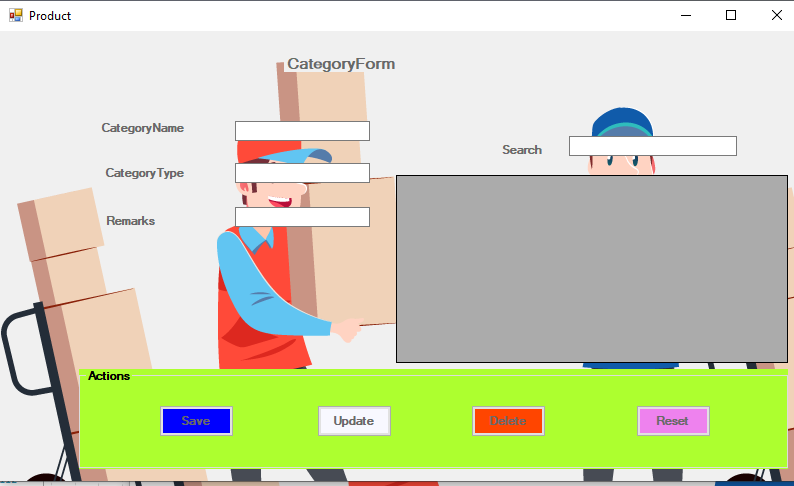






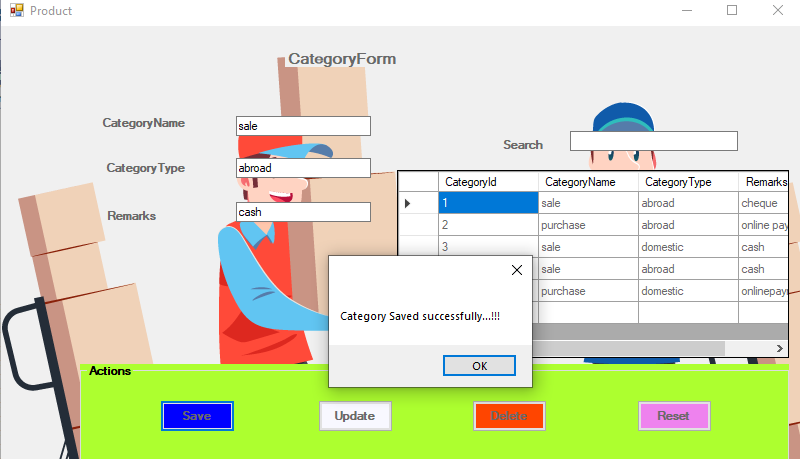


Category Form:

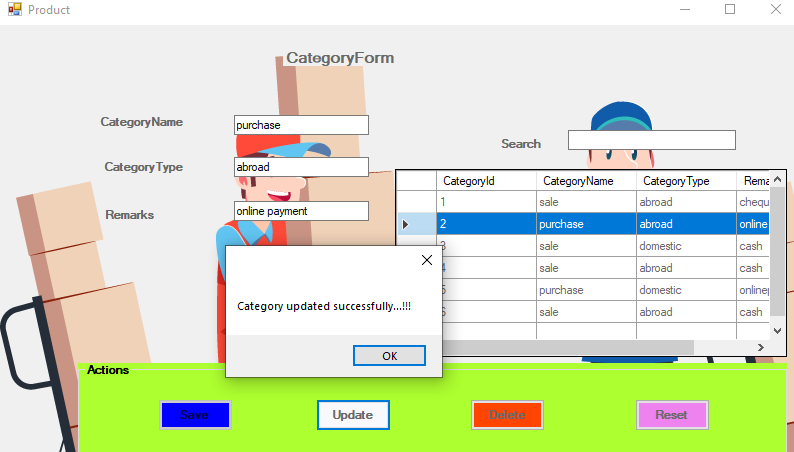


Save, Delete, Update, Reset operations:

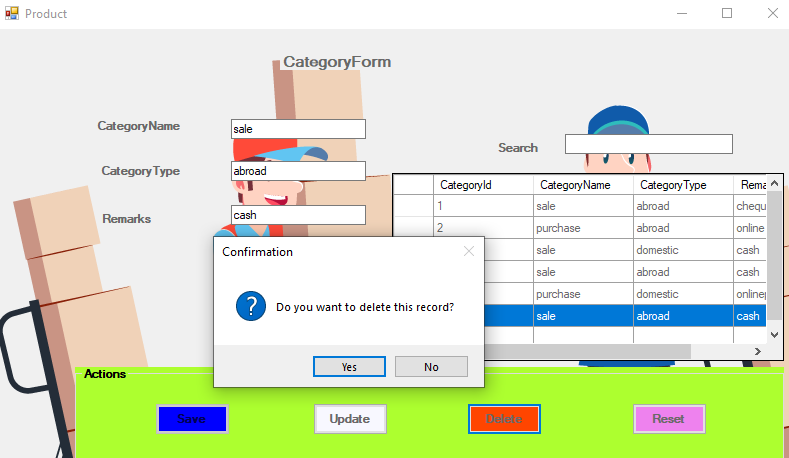
Save button:



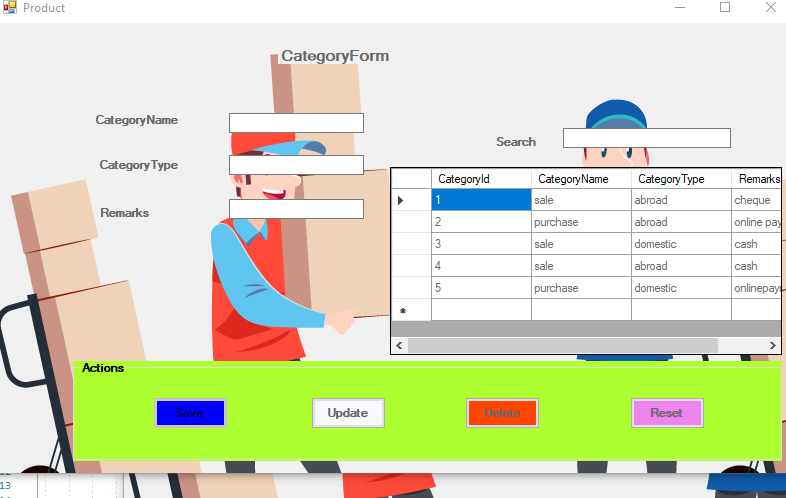
Update:



Delete:



Reset:



Code of Category form:







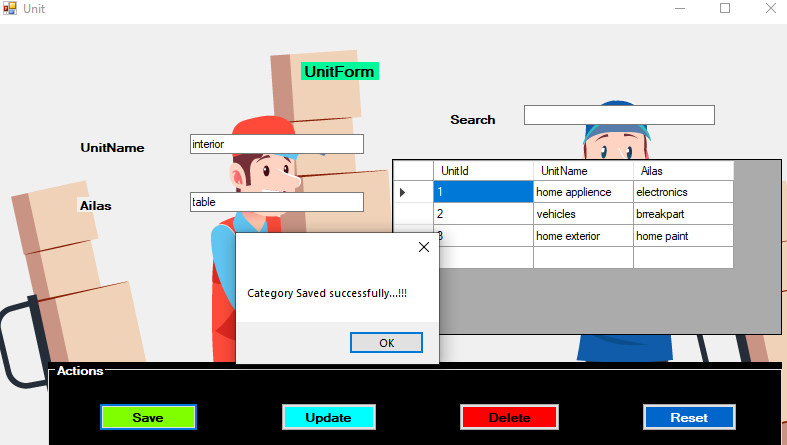


Unit form:

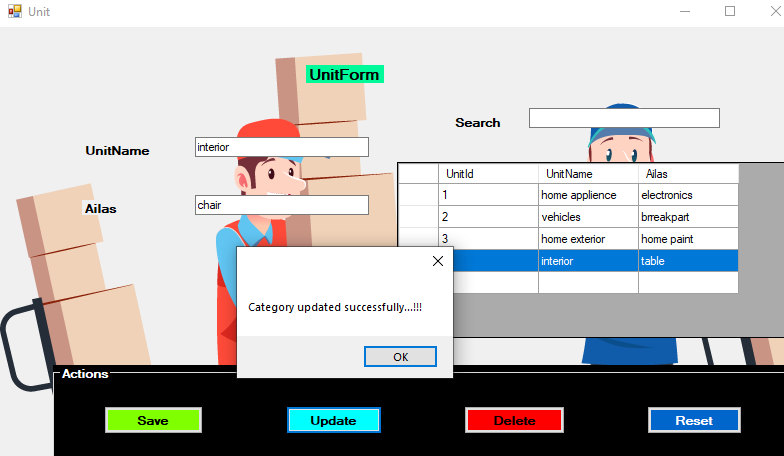


CRUD operations:

Save:



Update:



Delete:



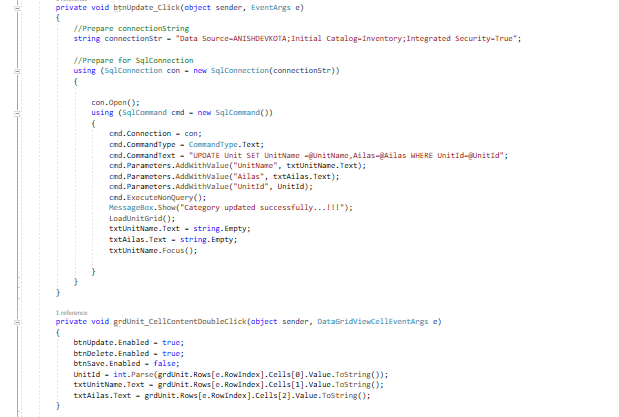
Reset:



Code of Unit form:

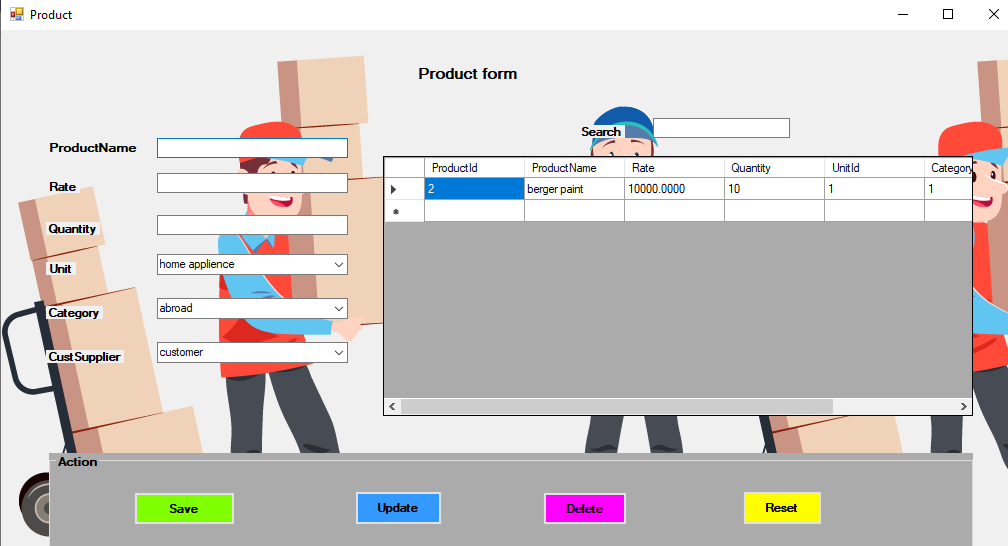






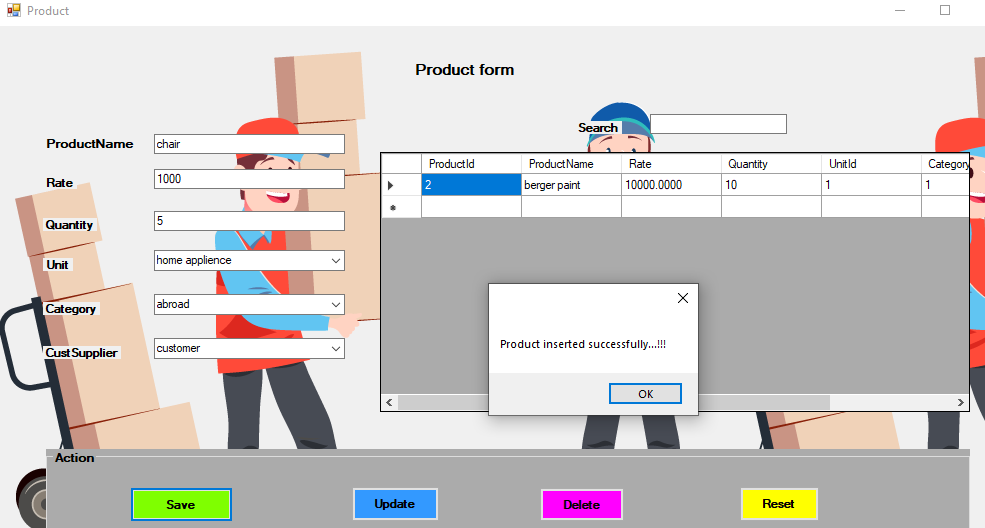


Product Form:

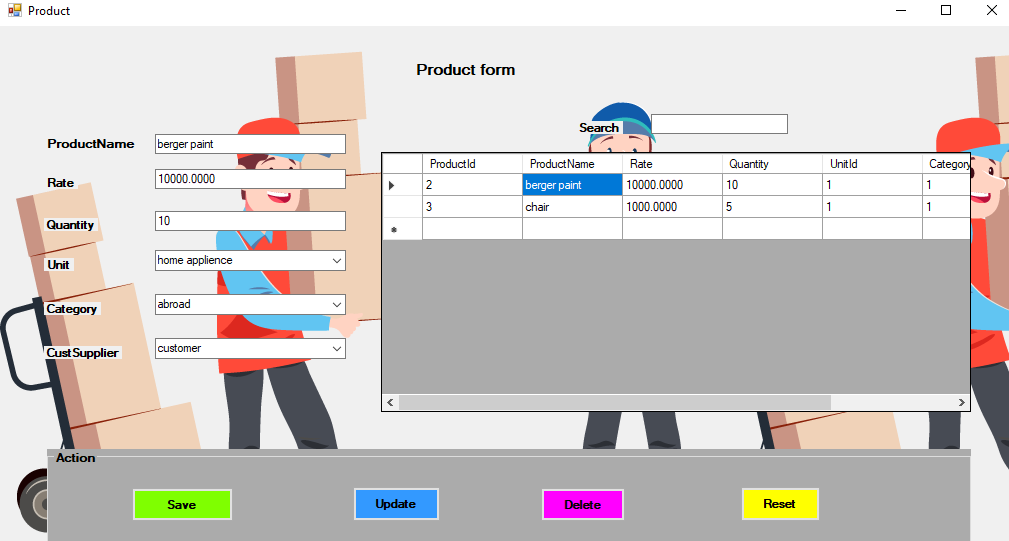


CRUD operation:

Save:



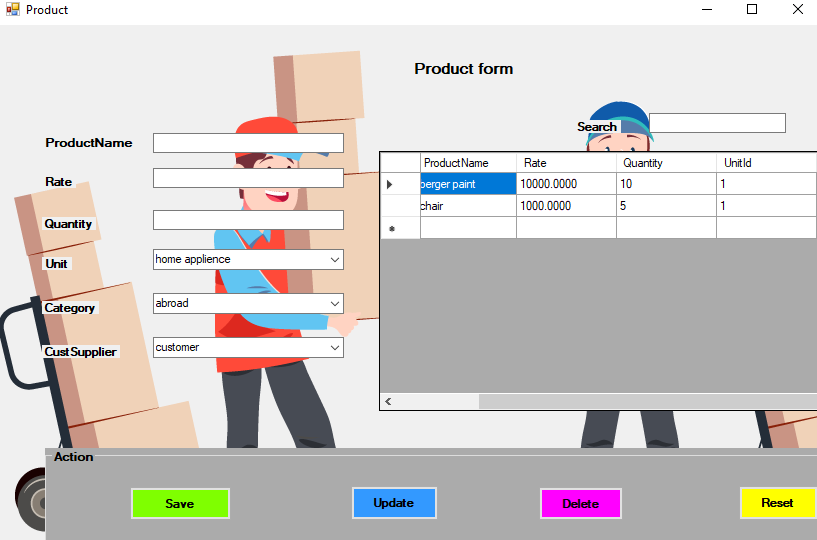
Update:



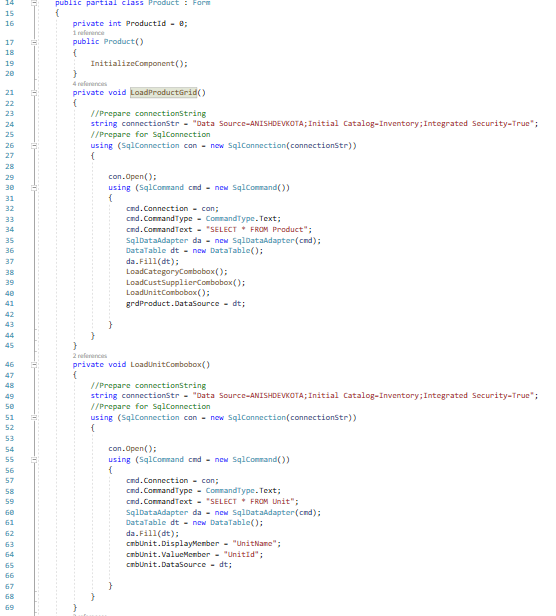
Delete:



Reset:



Code of Product form:



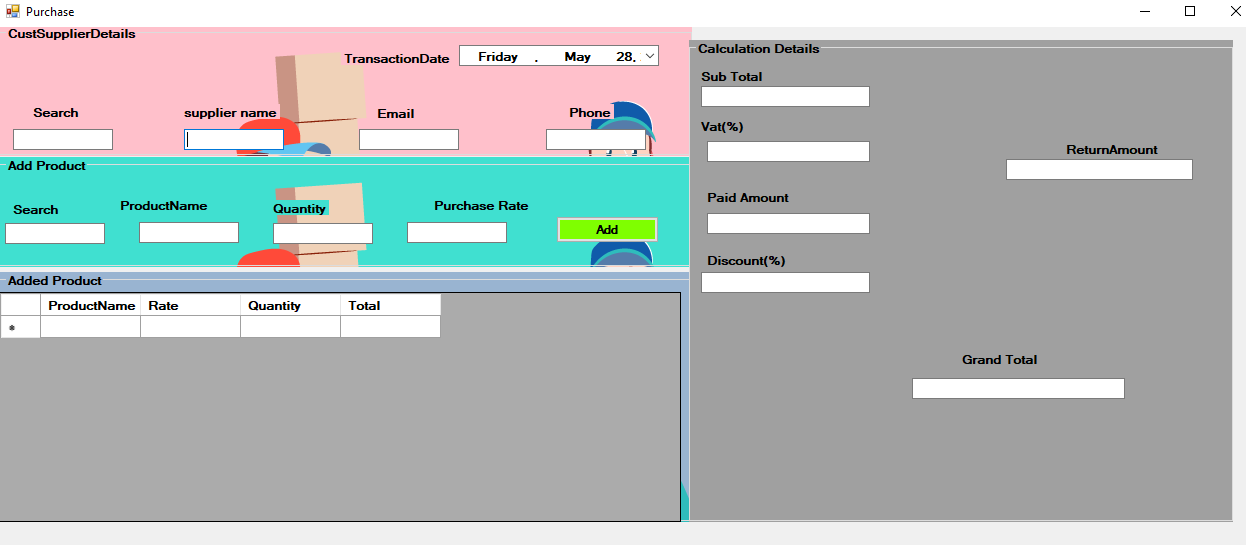


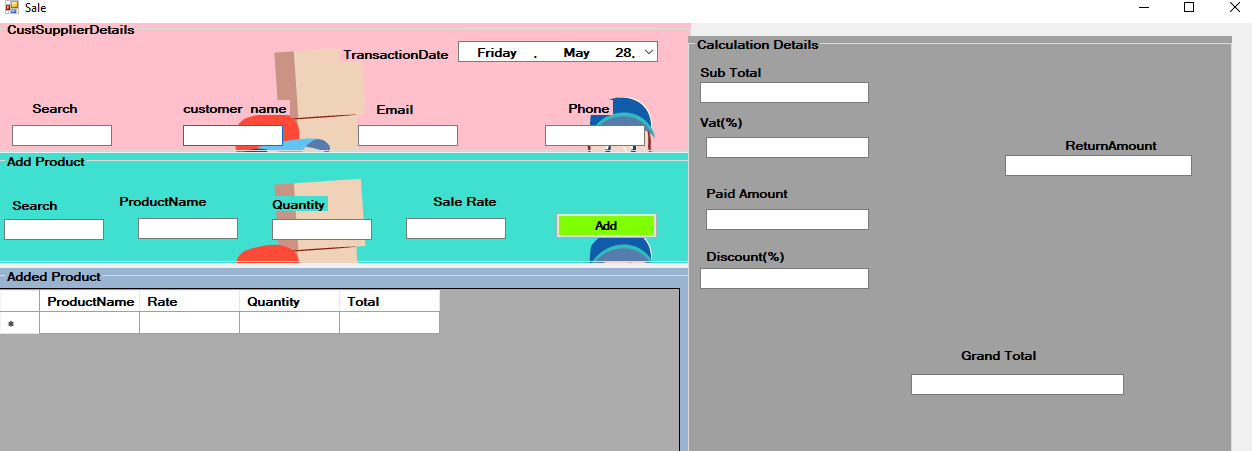




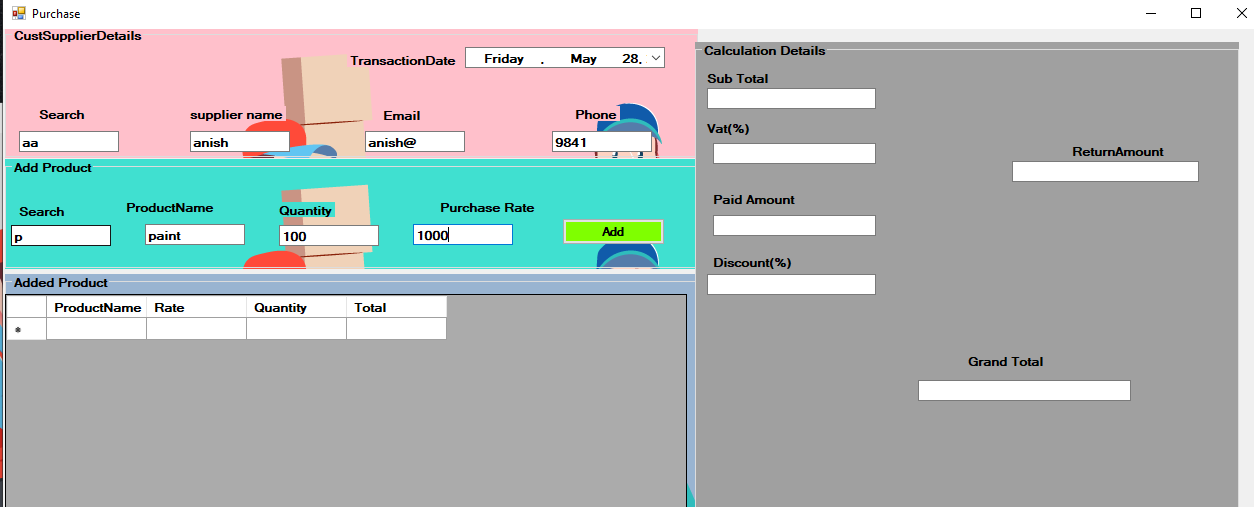


Purchase and sale Form:

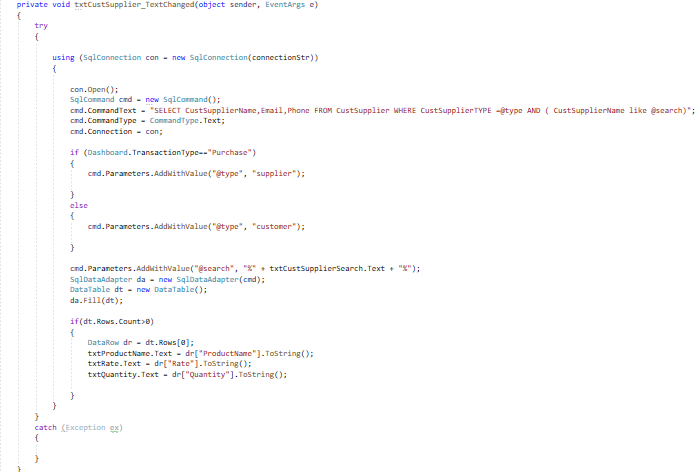




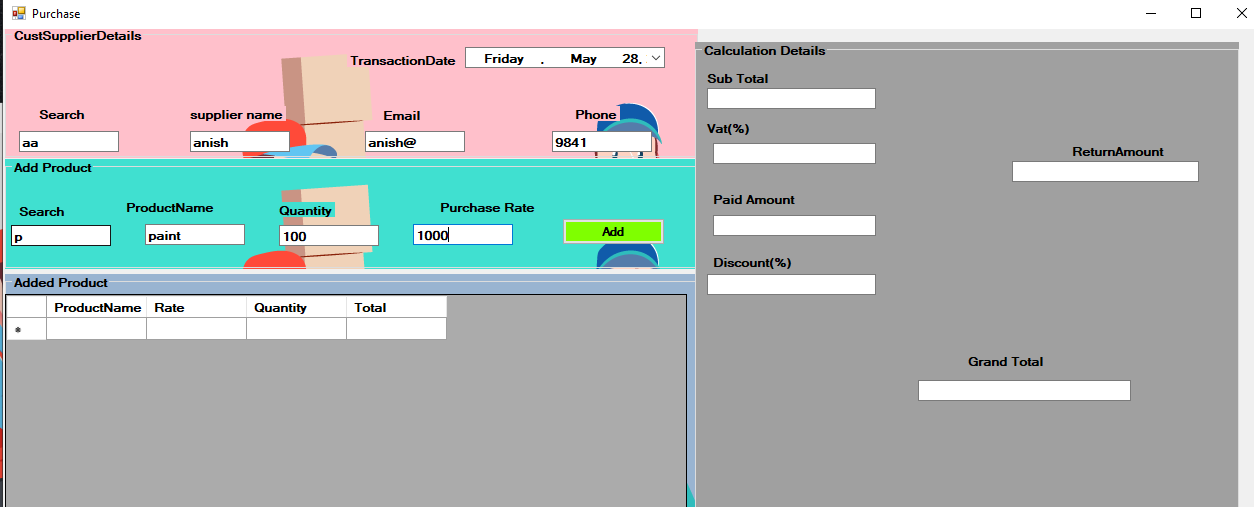
Search Customer Supplier:



Code:



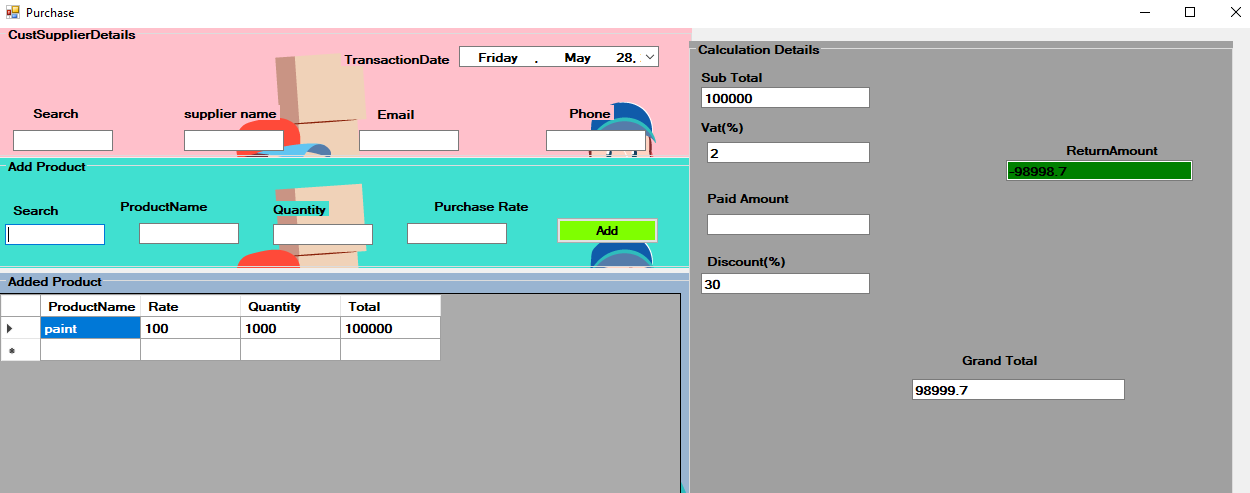
Search Product:



Code;



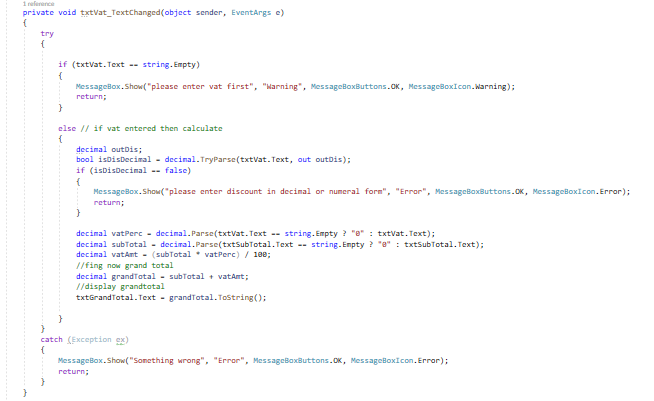
Add product with calculation:

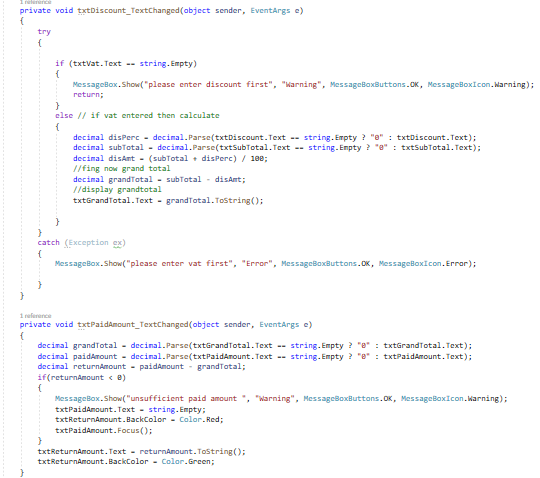


Code for Purchase and Sale:









Therefore, above are the snapshots of code and forms in our application.

# **IDE to manage the development process**

As an IDE application we have already selected the Visual Studio 19 for development process. Above we have also described the tools of IDE application. Some use of tools of Visual Studio 19 used during the development process are:

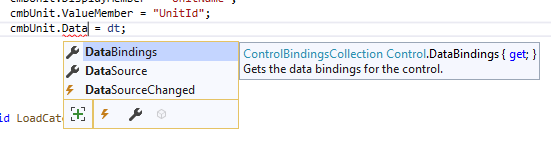
## **Code editor**

This is the tool of IDE where the main source code is written. This tool is designed for writing and editing the source code. Components of code editor with their justification and screen shot are given below:

1. Auto complete:

complete is a type of code editor that helps to reduce the time of writing code by suggesting or predicting text. While building our inventory management system, auto completion process advanced by IDE helped us by completing the statement code automatically. Due to this process user written code was minimized and system code was implemented which helped our application to be more accurate and convenient.

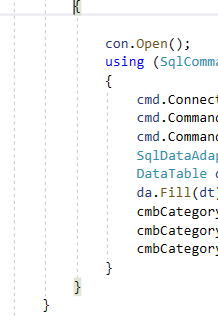
Screenshot of auto complete in Visual Studio 19:



1. Bracket matching:

Bracket matching is an editor that helps to determine the enclosing of certain code this also helps in finding the missing bracket and block of code. This process helped in making a block of code as a single statement. Because of this bracket we were able to determine where the code is mistaken in a block of code. This process also helped in enclosing a statements or code in a bracket.

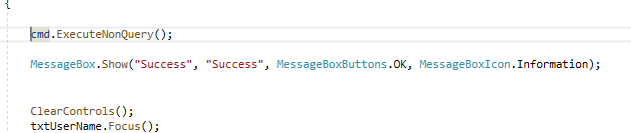
Screenshot of bracket matching in Visual Studio 19:



1. System check:

Syntax check also act as editor by recognizing the correct syntax and highlight the error in program. Because of this process we were able to detect the syntax which were wrong and this helped us in debugging our code. With the help of message box provides by Visual studio we implemented the code of showing invalid for incorrect user or password and successful for correct ones.

Screenshot of system check in Visual Studio 19:



## **Compiler**:

Compiling is a tool used by Visual Studio 19 mainly for documenting the source code and translate to computer in machine language. Likewise, compiler helped us by translating the written code into computer understanding language. Because of this specialty, our source text was delivered to the computer.

## **Documentation:**

Documentation tools in Visual Studio 19 help to generate a documented data extracted from the database. This tool also helps to describe tables and columns and database objects. Because of this documentation process of Visual Studio, we were able to save our codes and syntax of application. Also due to this we were able to access the saved file whenever needed.

## **Libraries**:

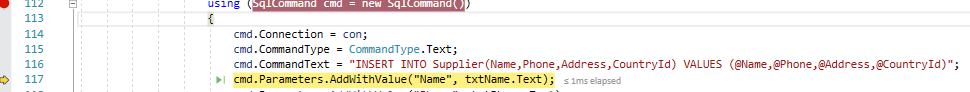
These tool provided us with functionality of providing the functions not included in a core programming language.

## **Build automatic:**

Automatic tools are programs that automate the creation of executable applications from source code. While establishing our Inventory System software in Visual Studio 19, automatic build helped us by executing the application according to the source code given by us.

## **Debug**:

Debugging means to find the error while giving the source code. When we run our application, debugging helped us by identifying the exact error location and solution to be implemented for it. Debugging process included in our project in Visual Studio:



# **Constraints of visual studio:**

We know that visual studio is one of the IDE software and it has a lot of features hence, it is very heavy, it’s impossible to use this program in a computer with low resources. Probably, a lightweight version should be created.  And because it is very heavy software with a lot of features going on inside it, it takes a lot of time to install it in the computer. The application in this software additionally used the large amount of space in computer.

# **Conclusion**

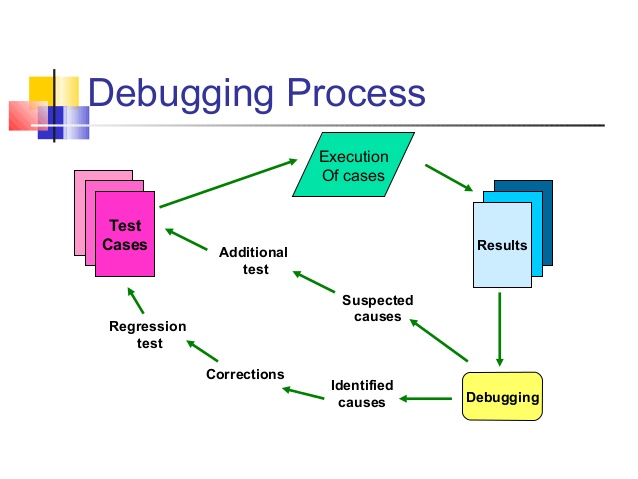
Lastly Visual Studio 19 an IDE application was used for implementing our application. Having more features and facility, our application development process was built successfully. And following the source code we were able to execute the final application. Therefore, we scripted our Inventory System application via C# and with the support of IDE application (Visual Studio)

# **Debugging**

Debugging process in Visual Studio 19 is done by identifying the problem stepwise, isolating the source of problem and correcting the problem. The process of debugging is to find the bugs and errors. Here debugging refers to check whether the code syntax is correct or not for processing. Debugging is the most important tools available in IDE that helps a lot for the programmer to build errorless program. It is an important procedure in any new programming or equipment improvement process. Talking about debugging, our team went through it in every source code of individual forms. This process helped us by finding out the error code and finalizing the accurate inventory system application (Definition of 'Debugging', 2021).

# **Process of debugging**

Debugging process starts firstly from finding out the error or bug in a source code by checking if the expected output in displayed or not. After finding the error in code, location must be marked out. Then analyzing the error and documents must be checked. After analyzing, correction is required in the wrong syntax. Being corrected source code is again tested by the developers for achieving the expected correct result.



The above debugging process is characterized as:

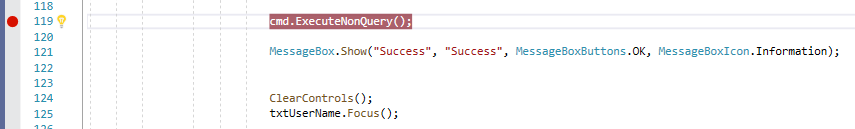
* Figuring out the possible errors
* Finding out the bug/error
* Analyze the possible correction
* Fixing the bug
* Confirmation for successful execution in IDE

# **Debugging facilities of IDE**

The IDE application we used also provided us the best facility of debugging our application. After finding the error and correcting self we tested for the main debugging process by IDE debugging in Visual Studio 19. Some major steps taken by IDE for testing our application program during our project were

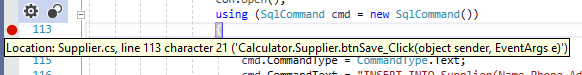
1. Breakpoint

Breakpoint means a certain place kept in our programming code which pause the execution of program and returning to the code for finding out the main error which have been occurred. Example of break point used in our programming code of application is shown below:



1. Tracer

This functionality in IDE produced a message for monitoring the implementation of program. race breakpoint has the diamond shape in the code editor gutter. Both a condition and a trace action can be specified on a breakpoint.

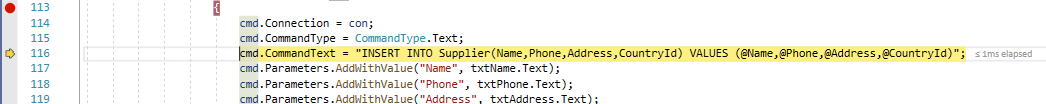


1. Watch point

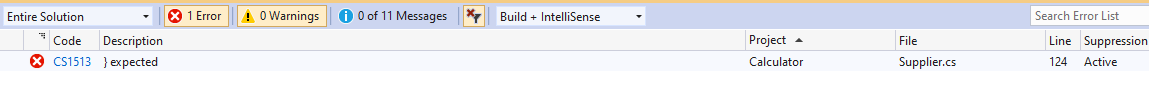
This process is related to the COBOL data item. Watch point is special type of breakpoint which pause the execution of program when the value of given field is changed without defining.

1. Step command

The step command functions on stepping into calls that were compiled. This process in debugging also helps for skipping the correct step and running the code for final output.



1. Figuring of bug:



# **Debugging process evaluation:**

We have already learned that debugging is the process of finding out the bug and coming with the required solutions for successful development. While development of application there are many bugs and errors encountered and in order to prevent from those disturbing errors, debugging process is used. In application development debugging process is required most for successful operation of any program.

Similarly, we also implemented the debugging process for our Inventory System application. While debugging we found lots of error and bugs and implemented the possible correction for the process. Our used IDE application (Visual Studio 19) also provided us the debugging tools which helped in finishing our task affectively. Because of the tools provided by Visual Studio our debugging task was done faster and efficiently. After using the tool of debugging we were able to find the exact error or problem that was affecting our application. Also due to suggestion for the encountered bug or syntax, Visual studio helped us analyze the real accurate syntax. With the help of debugging process by Visual Studio we finally invented our Inventory System application.

Figuring out the possible errors, finding out the bug/error, analyzing the possible correction, fixing the bug, Confirmation for successful execution were the functionality provided by Visual Studio 19. Some snapshots of the debugging process provided by our used IDE application in our Inventory System are shown below:

# **Coding standard**

Coding standard refers to the style of programming or source code written for building a successful application. Coding standard is also referred as well-defined and standard style of coding written for implementation of program. This is also known to be the path or method any developers follow for building a program and for knowing the code or each another executed. This makes a certain project in a team systematic and understandable.

Some major aspects of Coding Standard are:

* Naming of the file
* Formation of file or codes
* Convention naming while writing source code and saving it
* Testing process

# **Benefits of code standard:**

* It increases the efficiency and reduces the time for development
* It improves the maintainability and readability of code.
* It helps in maintaining coding consistency for the developers
* Reusable code and easy detection of error
* It provides the software security from the beginning

# **Constraints of code standard:**

* Developers or every individual member of team must work with the old implemented coding.
* Problem for new members if unknown to the followed standard in a team.

Coding standard helps developer individually by making the format of source code correct and systematic. Different syntax or code will be executed with same naming convention, variable declaration as we can say with the user comfort language. This helps developer to understand the code of different references and make the task documented clearly. Also this helps a developer for saving time because of code reusability.

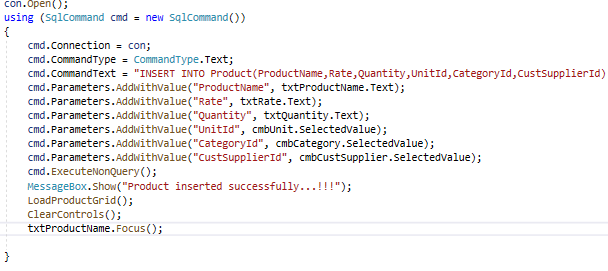
Coding system for team also helps in a team by making the rule of team to follow the similar and well known coding for project development. Following of same coding standard in team helps each other to understand the source very accurately. Editing also will be same because of having same name convention, brackets, testing etc.

# **Coding standard followed while delivering our application**

Our Inventory System application was developed within a team. Our team implemented the same coding standard for the entire development process. Example of coding standard with their screenshot while building our application are given below:

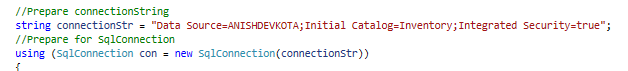
## **Indentation**

This is a type of coding standard whose main purpose is to make a source code or syntax readable and understandable. This method also helped for successful delivering of our final application. This method helped us by making our team to understand, read, modify, enhance the code easily. Below is the indentation process of coding standard followed by our team.



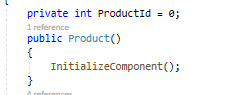
## **Commenting code**

This is the process of standard coding that helps in identifying source code as well as algorithm used in program. This is determined to be best before writing the actual code. Because of this, our team members were able to find the meaning of each algorithms and methods done in the form of coding.



## **Declaration of variable**

Variable declaration is also one of the component of standardized coding. As per following the code standard, our team declared the known data type and class for reusing it in other required parts of our application code. Variable consists of name and its characters. This method in coding helped us by knowing the name of variable and datatype it holds.



## **Naming convention**

It is a process of implementing the name to any function, variable, files and other entities throughout the application coding process. This is also a part of standard coding, which helps in writing a code following the valid naming method. Naming convention also helps in reducing effort to understand source code. Types of naming convention followed in a standard coding are:

## **Pascal naming convention**

This is a type of naming convention used for delivering the source code for application. This convention follows the naming rule of first letter of each word must be capitalized. This language is found to be widely used by software developers for writing source code to name functions, classes and objects.

Example: AnishDevkota

## **Camel naming convention**

This is a type of naming process where first letter of any word is written in small and the joining word is written in capital. This letter as by name also follows the shape of Camel for writing code. Most of the developers are also found to be using this convention.

Example: anishDevkota

Naming convention used by our team :



Hereby, for our Inventory System we have used the Pascal naming convention. Being implemented and having good idea in it, our team decided to follow camel case naming as our coding standard. And for the entire process of writing code, camel case was used for our development process.

# **Recommendation of coding standard**

Following coding standard rules are optional in not a compulsory factor for application development. But following this standard helps us to reduce the effort for our application development. This also helps in documenting an application code systematically. This standard makes the flow of any work faster and efficient.

So, personally talking about the standard we will recommend any team or individual to follow it. Due to this our development process went easily, coding was really understandable to each other, editing or making changes was also found to be easy by this standard. This method worked very well in debugging process and reusability of code was very functioning. Also mainly, our security was maintained well by following this principle. Although, it would be hard for the new ones for understanding the coding process whom are unfamiliar to the used code. Also implementing new languages or rules will be problematic while developing application.

At last, personally we would recommend for using this principle of coding for making reusable and understandable code and application while working in team as well as individual.

# **Conclusion**

Lastly, our team developed a successful inventory system according to the requirement of owner. Implementing the algorithm and valid programming language we finished in making the application. Above we also have given the evidences of our application in the form of snapshot. Also valid code and tools for developing are described above. Debugging, Coding principles, IDE tools, platform practiced by our team while writing code and designing application helped us very much for completion process and fixed the errors occurred problems in our application.

# **References**

Bolton, D. (2019, july 3). *ThoughtCo*. Retrieved from thoughtco.com: https://www.thoughtco.com/source-code-definition-958200#:~:text=Source%20code%20is%20the%20list,computer%20can%20understand%20and%20execut

*Computer Hope*. (n.d.). Retrieved from computerhope.com: https://www.computerhope.com/jargon/p/programming-language.htm

Definition of 'Debugging'. (2021). *The Economics Times*.

Doherty, E. (2020, 4 15). *educative*. Retrieved from educative.io: https://www.educative.io/blog/object-oriented-programming

*edgetechacademy*. (n.d.). Retrieved from edgetechacademy.edu: https://www.edgetechacademy.edu/node-js/event-driven-programming

*guru99*. (n.d.). Retrieved from guru99.com: https://www.guru99.com/c-pointers.html

Pedamkar, P. (n.d.). *EDUCBA*. Retrieved from educba.com: https://www.educba.com/algorithm-in-programming/

*sanfoundary*. (n.d.). Retrieved from sanfoundary.com: https://www.sanfoundry.com/java-program-find-simple-interest/

*sitepoint*. (n.d.). Retrieved from sitepoint.com: https://www.sitepoint.com/control-statements-in-c-part-1/

*technopedia*. (n.d.). Retrieved from technopedia.com: https://www.techopedia.com/definition/8982/procedural-language

*ThinkAutomation*. (n.d.). Retrieved from thinkautomation.com: https://www.thinkautomation.com/eli5/what-is-an-algorithm-an-in-a-nutshell-explanation/

*tutorialspoint*. (n.d.). Retrieved from tutorialspoint.com: https://www.tutorialspoint.com/csharp/csharp\_environment\_setup.htm

*veracode*. (n.d.). Retrieved from veracode.com: https://www.veracode.com/security/integrated-development-environment