**A close up of a sign

Description automatically generated**

**AMERICAN INTERNATIONAL**

**UNIVERSITY-BANGLADESH** 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: **Asmaul Hossain Akash**

Student ID: **20-44209-3**

Program**: BSc (CSE)**

Semester: **Fall 20-21**

Section: **B1**

Submission Date: **02 January 2021**

Assignment Title: **Report on Lab Task 10**

Assignment No: **Final Term Assignment on Lab task 10**

Course Title: **Introduction to Programming**

Course Code: **00152**

Course Teacher: **Aneem Al Ahsan Rupai**

**Universal Billing System**

**Introduction:**

This code, Universal Billing System, is basically a desktop application which is developed in C++ platform. For developing a code, this C++ report will also provide basic guides for the user. This report and application's main aim is to provide an application that will be helpful for any departmental stores, shops, cafes, accessories shop, etc. Instead, these stores and shops will do the billing to the customers with this application's help.

**Object Oriented Principles-**

1. **Encapsulation**:

Encapsulation is an Object-Oriented Programming concept that binds together the data and functions that manipulate the data, and that keeps both safe from outside interference and misuse.  It is one of the popular feature of Object Oriented Programming that helps in data hiding.

1. **Inheritance:**

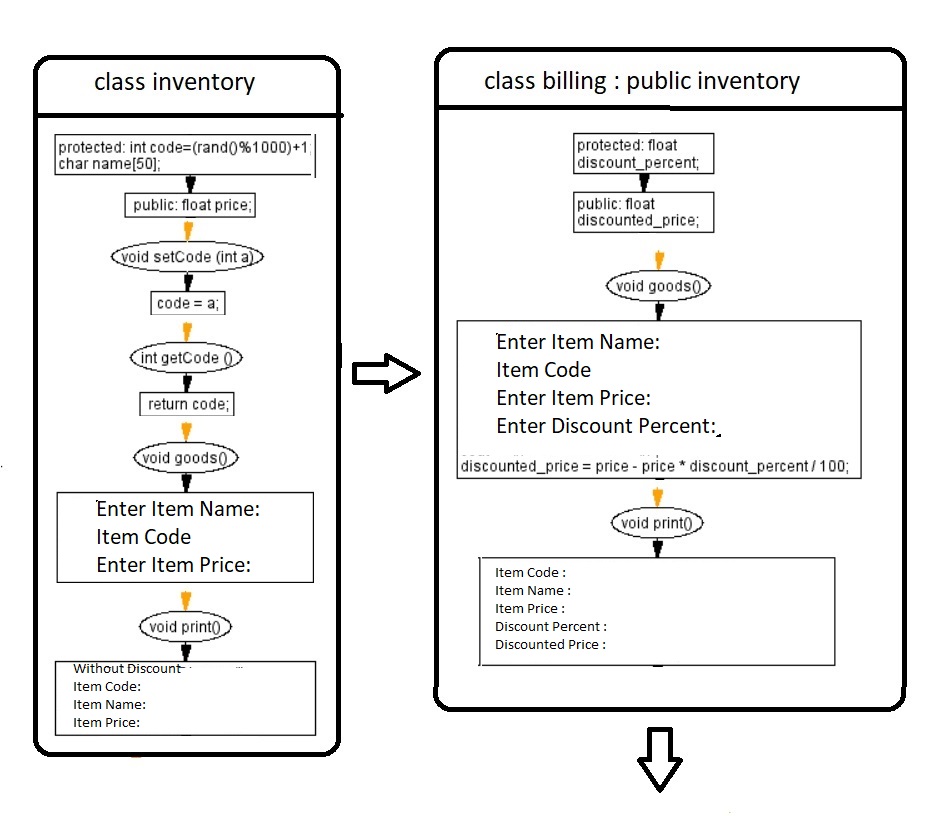
Inheritance is a mechanism in which one class acquires the property of another class. The class whose properties and methods are inherited is known as the Parent class. And the class that inherits the properties from the parent class is the Child class. In such way, one can reuse, extend, or modify the attributes and behaviors which are defined in parent class.

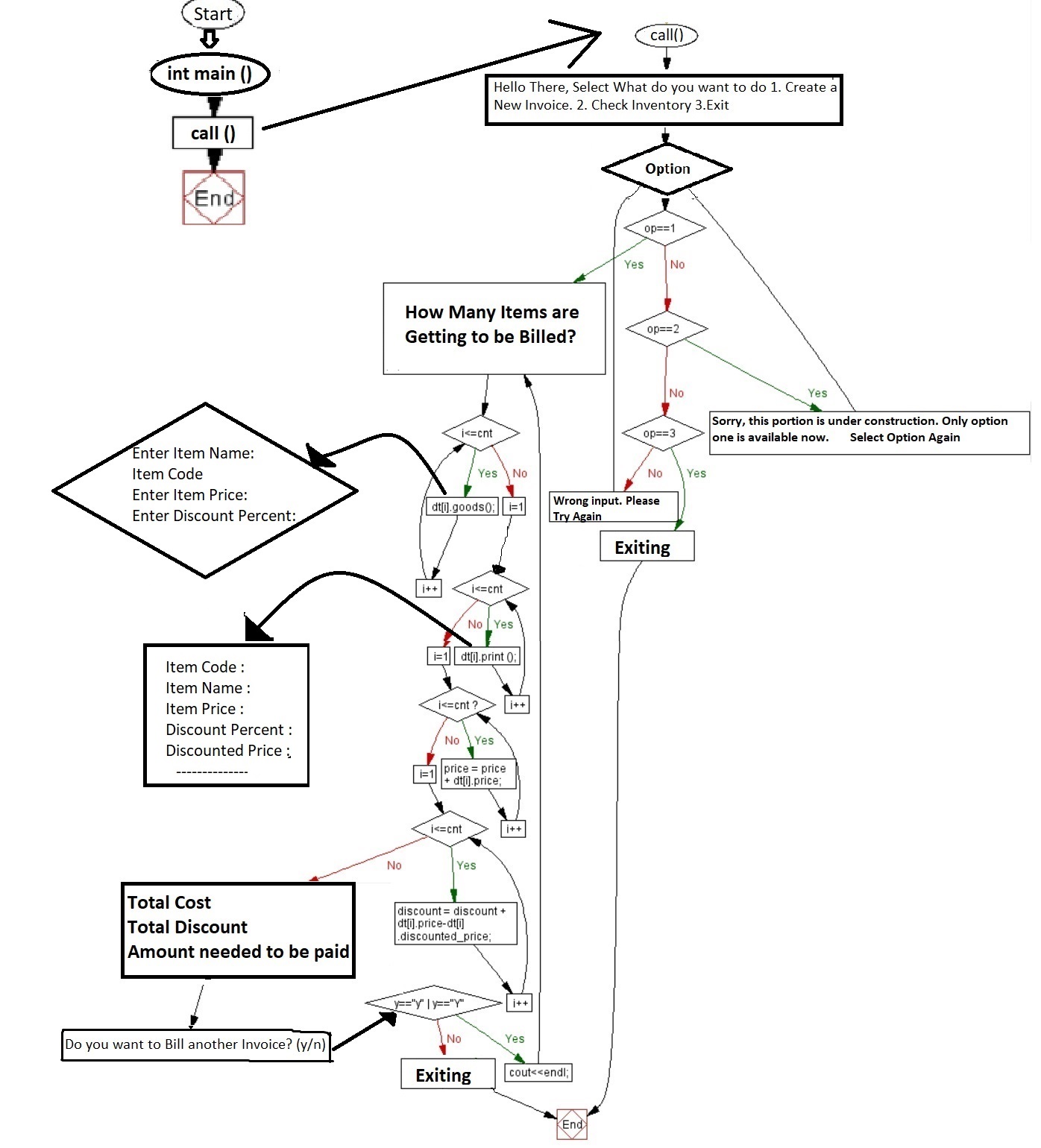
1. **Polymorphism:**

The word polymorphism means having many forms. In simple words, polymorphism can be defined as the ability of a message to be displayed in more than one form. polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function. In C++ polymorphism is mainly divided into two types: Compile time Polymorphism and Runtime Polymorphism.

**How my system works-**

**1. Flowchart**





**2. Description of your system**

1. **Why have you developed it?**

**Ans:** The main propose of this billing system application is developing a system that automate the bill creation. While many products are sold in a shop, bill submission is very tiresome work and maintaining the record of bill is very difficult and time-consuming. In the present system, the staff has to work manually to create an invoice, and it is tough to calculate the final cost if there is some discount on individual products. Sometimes in the manual process, there is a possibility to get errors. To overcome all these problems, I have developed this bill management system.

1. **What facilities it provides?**

**Ans:** This application is developed to manage the billing process. Using this, a staff can create invoice easily. If there is a percentage of discount in individual product, the creation of bill seems complicated. With this application, employees can easily create invoices even if there's a percentage of discount in specific products. It also shows the actual cost and net-discounted price.

1. **How it works?**

After the code runs by user, first main function will call call() function. In call () function it will show starting message and will ask for take input of options available. If the user presses 3, then the system will select option 3 “Exit” and end the process.

If the user selects option 2, the system will show a message and ask input options again. If the user selects option 1, then the system will go forward. And will ask for the quantity of products.

After inputting the quantity, the system will create object with array in the backend process. Then the system will call goods() function from billing class. For the goods() function, the system will ask to input item name, item price and discount percentage. Item code will generate randomly.

After taking all these inputs for all products, the system will calculate the discounted price for individual products and call print() function from the billing class. For the print() function, the system will show Item code, name, discount percentage and discounted price for individual products. The system will then calculate the net price by summing all product prices and calculate the net discount price. Finally, the system will show the final outputs: total cost, total discount, and amount needed to pay by customer. After that system will ask if the user wants to create another invoice. If the answer is Yes, then the system will go to the option one again, and if the answer is No, then the system will end the process.

**Use of OOP within the system-**

In the system implementation, I used three OOP Features.

**Encapsulation**: There are two encapsulations in the system. First is in inventory class - protecting data (code, name) by declaring them as protected access modifier. Second one is in billing class. There is one data (discount\_percent) in the protected access modifier.

**Inheritance:** In the system, billing class inherits inventory class and extends by adding 2 extra data members (discount\_percent and discounted\_price) in the class.

**Polymorphism:** There is two polymorphism behaviors in the system. Both are runtime polymorphism also as known as function overriding. In parent class there is goods() and print () function which have some data members. Moreover, there are the same goods() and print () function in child class, but there are more data members in the function in child class.

**Use of Modularity in my system-**

There is two class and many modules in these class. Now I will be writing about these.

1. setCode() module, which is used to set up "code" data (item code).
2. getCode() module, which is used to return the value which is set in setCode module.
3. goods() module, which is created to take input of items, price in parent class and extended with discount percentage in child class.
4. print() module, which is created to show the output of Item code, name, price in parent class and extended with discount percentage and discounted price in the child class.
5. call() module, which is created to print first output, calculate the bill and print the final outputs.
6. main() module, which is the main function of the system.