NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT
ON

WORKOUTS SCHEDULER

Submitted in partial fulfillment for the award of the degree of Bachelor of Engineering

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

SUJAY HAZRA

1NH17CS127

IV-B SECTION

Reviewed by

Ms. Priti B. Badar
(Sr. Assistant Professor,CSE)

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

Certificate

This is to certify that the mini project work titled

WORKOUTS SCHEDULER

Submitted in partial fulfillment for the award of the degree of Bachelor of Engineering

SUJAY HAZRA
1NH17CS127

During the academic year
2018-2019

Signature of Reviewer **Signature of HOD**

Semester End Examination

Name of the Examiner	*Signature with date*
1. ……	***……***
2. ……	***……***

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned my efforts with success.

I thank the management, **Dr. Mohan Manghnani**, Chairman of NEW HORIZON EDUCATIONAL INSTITUTIONS for providing necessary infrastructure and creating good environment.

I also record here the constant encouragement and facilities extended to me by **Dr. Manjunatha**, Principal, NHCE, **Dr. Prashanth.C.S.R,** Dean Academics, **Dr. B. Rajalakshmi**, Head of the Department of Computer Science and Engineering. I extend my sincere gratitude to them.

I express my gratitude to **Ms.Priti B. Badar,** Sr. Assistant Professor, my project reviewer for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Finally a note of thanks to all the teaching and non-teaching staff of Computer Science and Engineering Department for their cooperation extended to me and my friends, who helped me directly or indirectly in the course of the project work.

SUJAY HAZRA

1NH17CS127

ABSTRACT

It is very important for a trainee as well as the trainer to keep track of workouts done in the previous sessions as well as to get an insight on an upcoming session, as a Gym goer myself, I have faced a similar problem and found that the best probable solution would be for a program to keep track of workout routines, make workout plans for the week and remind me about the workouts to be done on the current day and other such features that would benefit any trainee. This program is developed to solve the above problem.

This program can be cited in the following steps:-

1) User Enters his workout routine.
2) Current workout is displayed and user can select if they have completed the workout
3) Uncompleted workouts are displayed as missed workouts.
4) The schedule can be altered and rescheduled in an user friendly way.

Further improvements can be made by making the program online, Higher-end Graphics can be provided making it a GUI, workouts can be displayed with tutorials, multiple schedules can be managed, msg alerts can be added if workout is missed, etc.

TABLE OF CONTENTS

CHAPTERS

CHAPTER 1

INTRODUCTION

1.1 COURSE OBJECTIVES

The programming languages- C & C++ are one of the most widely used programming languages in the world. When compared to C++, C is a subset of C++ whereas C++ is a superset of C and can run most of C code while C cannot run C++ codes.

C being Procedural Programming Language does not support Object Oriented Programming; therefore it has no support for polymorphism, encapsulation and inheritance, while being an Object Oriented Programming language C++ supports all of the mentioned methods.

To sum it all up, the following features are present in C++ & absent in C:-

- Supports both function and operator overloading
- Functions can be used inside a structure
- Supports Reference Variables
- Supports virtual and friend functions
- Provides support for exception handling(error handling)

Hence, with all these features we can show the effectiveness and versatility of a program being written in C++ rather than in C. The Main Objective of this Mini Project is to understand Object Oriented programming concepts and techniques such as polymorphism, inheritance, data encapsulation, and various other oops concepts in a depth, develop problem solving ability and gain debugging skills. and implement them to solve day to day problems faced by people in general with the help of a program that is versatile, compatible and user friendly.

1.2 PROBLEM DEFINITION

During gym training, it is very important for a trainee as well as the trainer to keep track of workouts done in the previous sessions as well as to get an insight on an upcoming session, as a Gym goer myself, I have faced a similar problem and found that the best probable solution would be for a program to keep track of workout routines, make workout plans for the week and remind me about the workouts to be done on the current day and other such features that would benefit any trainee.To solve this problem we can use Object Oriented Programming concept to create a C++ Program to keep track of workout routines for an individual trainee. The Highlighted features being:-

*Displaying their next workout plan

*Notifying them incase they missed the plan

*User is allowed to schedule and make plans to their workout routines according to days specified(mon,tue,wed..)

.

1.3 OUTCOMES OF THE PROJECT WORK

This program can be used to create an application to be used by the trainee to help keep track of workout routines and facilitate reducing the bad habit of doing the wrong workout on the wrong day, the program can also add on many other features that would benefit any trainee to a greater extent.

CHAPTER 2

OBJECT ORIENTED PROGRAMMING CONCEPTS

2.1 CLASSES

Classes which are defined act as blueprints for objects, classes do not take up any memory space until an object of the type class is created. Data members of a class are the same for each object but their values are specific for different objects and operations performed on each object are same as the member functions do not differ for different objects of the same class. Once a class has been defined, any number of objects belonging to that class can be created corresponding to that class. Classes are user-defined and behave like built-in types of a programming language. In this project classes are used to give blueprints for the workouts consisting data members as workout names, days, etc.

2.2 OBJECTS

Objects are instances of a class, each object has its own values unique from other objects, but each object is of the same type class. For example, if there is a team of cricketers then each cricketer has different statistics but all the statistics are of the same type and are unique to each cricketer of a team. Objects in programs help show such differences. In this program workouts and days are objects.

2.3 DATA ABSTRACTION

The act of representing essential features of a program without including background details or explanation is known as data abstraction. This can be achieved through access specifiers such as private, public and protected provided in C++.

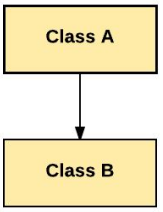
2.4 DATA ENCAPSULATION

Encapsulation is wrapping up of data and functions into a single unit. This is generally achieved through classes, the data in the classes is not accessible by any other part of the program directly without using objects, only the member functions of the class can access these data members and modify their values. This insulation of data from direct access by the program is called data hiding or information hiding.

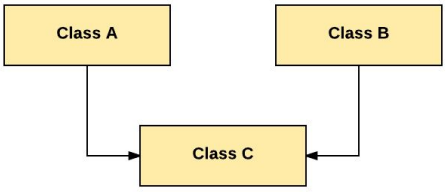
2.5 INHERITANCE

Inheritance is the process by which objects of one class known as the child class acquire the properties of another class known as the base class. It supports the concept of hierarchical classification.
The types of inheritance are: -

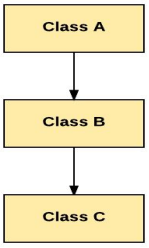
- Single Inheritance



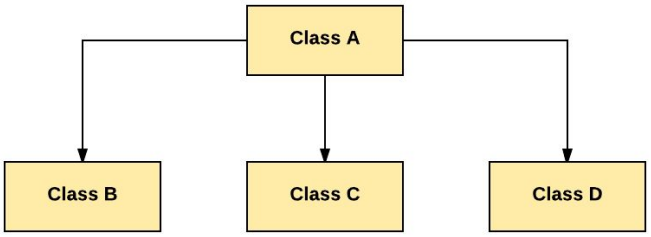
- Multiple Inheritance



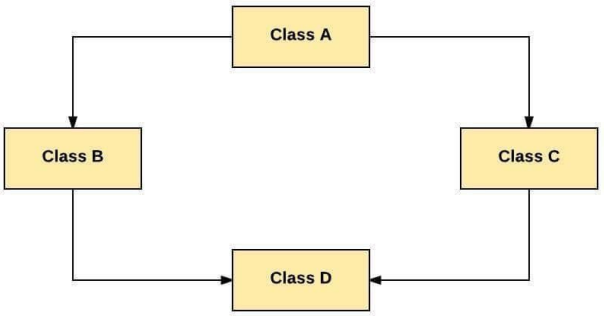
- Multilevel Inheritance



- Hierarchical Inheritance



- Hybrid inheritance



2.6 POLYMORPHISM

Polymorphism manifests itself by having multiple methods all with the same name, but slightly different
functionality.

There are two types of polymorphism: -

- Overriding
- Overloading
  - ❖ Function Overloading
  - ❖ Operator Overloading

2.7 DYNAMIC BINDING

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run time. It is associated with polymorphism and inheritance.

2.8 MESSAGE PASSING

An object-oriented program consists of a set of objects that communicate with each other.

The process of programming in an object-oriented language, involves the following basic steps: -

- Creating classes that define object and their behavior.
- Creating objects from class definitions.
- Establishing communication among objects.

CHAPTER 3

REQUIREMENTS AND DESIGN

Concept requirements :

To understand the basics of C++ concepts.

To understand the Object Oriented concepts.

3.1 Hardware Specification

Processor : Intel core i3 and above
Speed : 3.20GHz to 3.60 GHz
RAM : 512 MB and above
Hard disk : 256GB and above

3.2 Software Specification

Operating system : Windows 7 or above
(Any of the below)
Code blocks IDE
Dev C++
Turbo C++

3.3 ALGORITHM

Step 1- Start

Step 2- Creation of Classes

Step 3- Display the Main Menu

3.1- Start application and Enter Workouts Initializes the classes, and all the data related to the workouts are entered in a user friendly environment by the User.

3.2- Exit

Step 4

4.1- Display Current Workouts of the day

A list of workouts to be done on the current day is displayed as the user opens the application(According to system date).

4.2- Enter if workout completed

If the workout for the current day is completed the user has to mention this way next workout routines is showed and this way incomplete workouts can be tallied.

4.3- Display upcoming workouts

Gives a small hindsight on the workout schedule for the week

4.4- Display Missed Workouts

Displays incomplete or missed workouts.

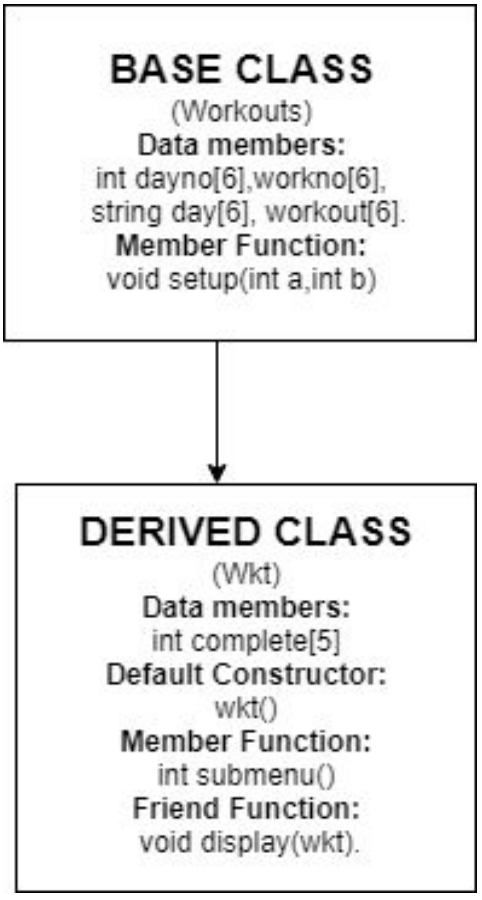
4.5- Edit Workouts(Reschedule)

Edit the list of workouts again

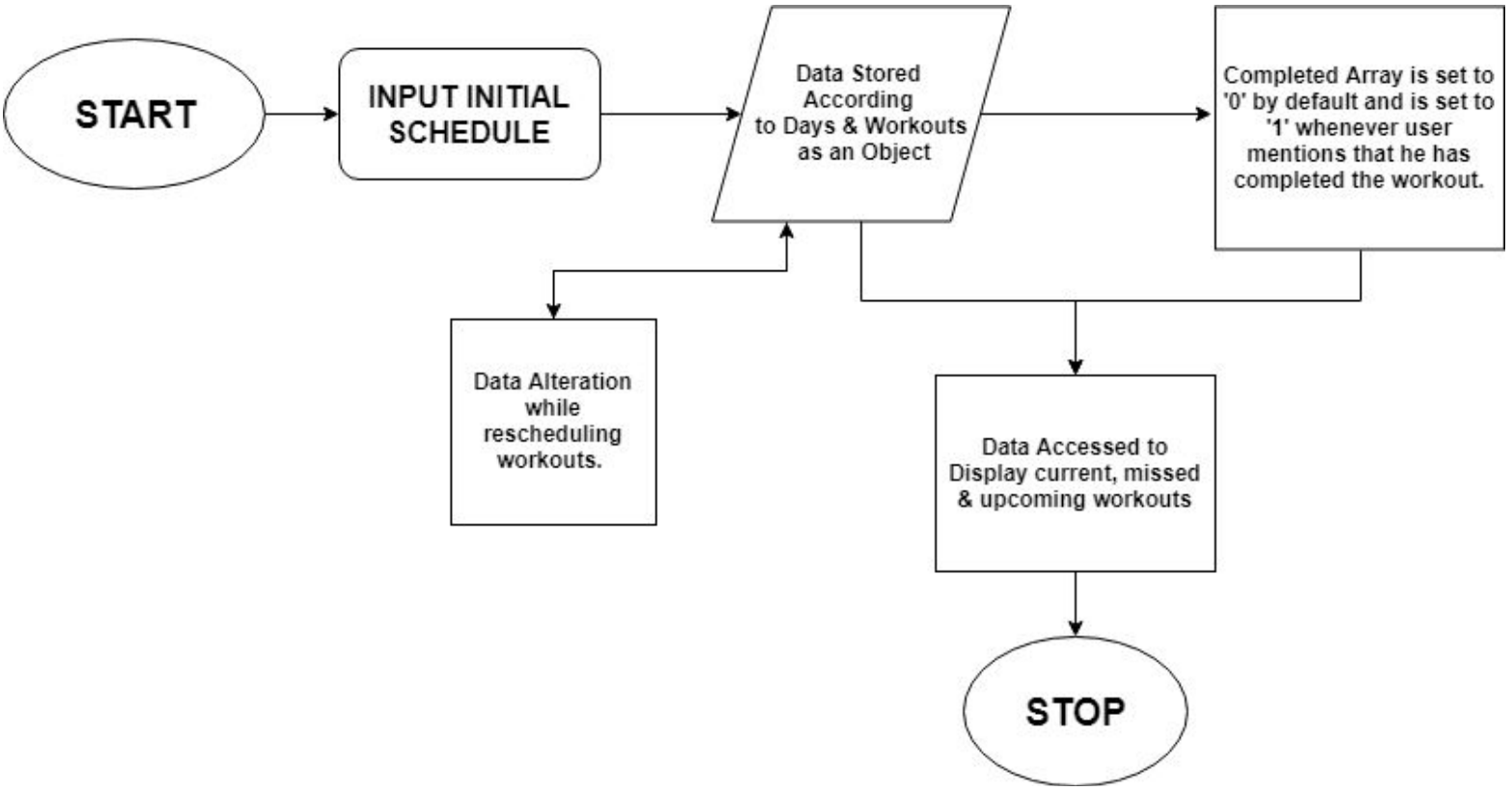
4.6- Exit

Step 5- Stop

3.4 CLASS DIAGRAM



3.5 DATA FLOW DIAGRAM



CHAPTER 4

IMPLEMENTATION

CLASSES USED:

1. **Base Class-Workouts**

```
class workouts
{
 public:
  int dayno[6],workno[6];
 string day[6], workout[6];
 public:
                void setup(int a,int b)
                {
                        a=a+1;
                        dayno[temp]=a;
                        workno[temp]=b;
                         if(a==1)
                         day[temp]="Monday";
                         if(a==2)
                         day[temp]="Tuesday";
                         if(a==3)
                         day[temp]="Wednesday";
                         if(a==4)
                         day[temp]="Thursday";
                         if(a==5)
                         day[temp]="Friday";
                         if(a==6)
                         day[temp]="Saturday";
                         if(b==1)
                         workout[temp]="Biceps";
                         if(b==2)
                         workout[temp]="Chest";
```

```
                if(b==3)
                workout[temp]="Shoulders";
                if(b==4)
                workout[temp]="Triceps";
                if(b==5)
                workout[temp]="Back";
                if(b==6)
                workout[temp]="Legs";
                temp++;
        }
};
```

The Base class has 4 data members - workout name, workout number, class name, class number and the member function setup() which is used to enter values to the data members of the object belonging to the class/ derived class.

2. Derived Class-Workouts

```
class wkt:public workouts
{
        int complete[5];
        public:
                        wkt()
                        {
                                for(int j=0;j<5;j++)
                                 {
                                  complete[j]=0;
                                 }
                         }
                        int submenu()
                         {
                           int a;
                         cout<<"Select An Option:\n1. Workout Completed(select 1 to mark if you
                        have completed current workout)\n2.Reschedule Workouts\n3.Exit"<<endl;
                           cin>>a;
```

```
if(a==1)
{
        if(complete[today-1]==0)
         {
                complete[today-1]=1;
                 return 1;
        }
        else
        {
                cout<<"Workout already complete"<<endl;
                getch();
                return 1;
        }
 }
        else if(a==3)
         exit(0);
        else if(a==2)
        {
          system("cls");
          cout<<"Displaying Current Schedule :"<<endl;
          cout<<"DAY:\t\tWORKOUT:"<<endl;
          for(int i=0;i<6;i++)
          {
                cout<<i+1<<"."<<day[i]<<"\t\t"<<workout[i]<<endl;
          }
          cout<<"Choose which day to be changed(1-6) *0 to
                    cancel"<<endl;
          cin>>temp;
          if(temp==0)
          return 1;
          temp=temp-1;
          cout<<"Choose Workout to replace previous one"<<endl;
          dispwk();
```

```
                cin>>a;
                setup(temp,a);
                cout<<"Successfully updated"<<endl;
                getch();
                return 1;
                }

    }
    friend void display(wkt);
};
                void display(wkt obu)
        {
                time_t theTime = time(NULL);
                struct tm *aTime = localtime(&theTime);
                int da = aTime->tm_mday;
                int month = aTime->tm_mon + 1; // Month is 0 - 11, add 1 to get a
                                jan-dec 1-12 concept
                int year = aTime->tm_year + 1900;
                today=dayofweek(da,month,year);
                system("cls");
                cout<<"Today is "<<obu.day[today-1]<<"\nYour workout is :
                        "<<obu.workout[today-1]<<endl;
                 if(obu.complete[today-1]==0)
                 cout<<"Workout not completed yet"<<endl;
                 else
                 cout<<"Workout Complete"<<endl;
                 cout<<"\n\nUPCOMING WORKOUT:\n";
                cout<<"Tomorrow is "<<obu.day[today]<<"\n\nWorkout for
                          tomorrow is : "<<obu.workout[today]<<endl;
                cout<<"\n\nMissed Workout:\n";
                for(int i=0;i<today-1;i++)
                {
                        if(obu.complete[i]==0)
```

```
                cout<<obu.day[i]<<" - "<<obu.workout[i]<<endl;
            }
        };
```

The class wkt is a derived class off the base class workout, it has a data member- int complete[5], ie an array to store an integer for each day to differentiate if the workout has been completed or not (1=complete, 0= incomplete).

Member functions are as follows:

1) Constructor (wkt())

   It is a default constructors which initializes the completed array to zero.

2) int submenu()

   This function gives the user the option to pick from after intitializing the schedule for the week, the user can now mark his current workout to be completed, reschedule his workouts or exit the application.

3) friend void display(wkt)

   This friend function is used to display the current workout after taking the system date automatically, it displays current schedule, missed workouts and upcoming workouts to the user.

MAIN FUNCTION:

```
int main()
{
 wkt ob;
 int x,l=1;
  cout<<"Welcome to Workouts Scheduler\n\nENTER YOUR WORKOUT SCHEDULE
      ACCORDING TO
DAYS\nDAY1-Monday\nDAY2-Tuesday\nDAY3-Wednesday\nDAY4-Thursday\nDAY5-Frida
      y\nDAY6-Saturday\n";
 for(int i=0;i<6;i++)
 {
  w: cout<<"\nChoose workout for Day"<<i+1<<endl;
  dispwk();
  cin>>x;
  if(x>0 && x<7)
```

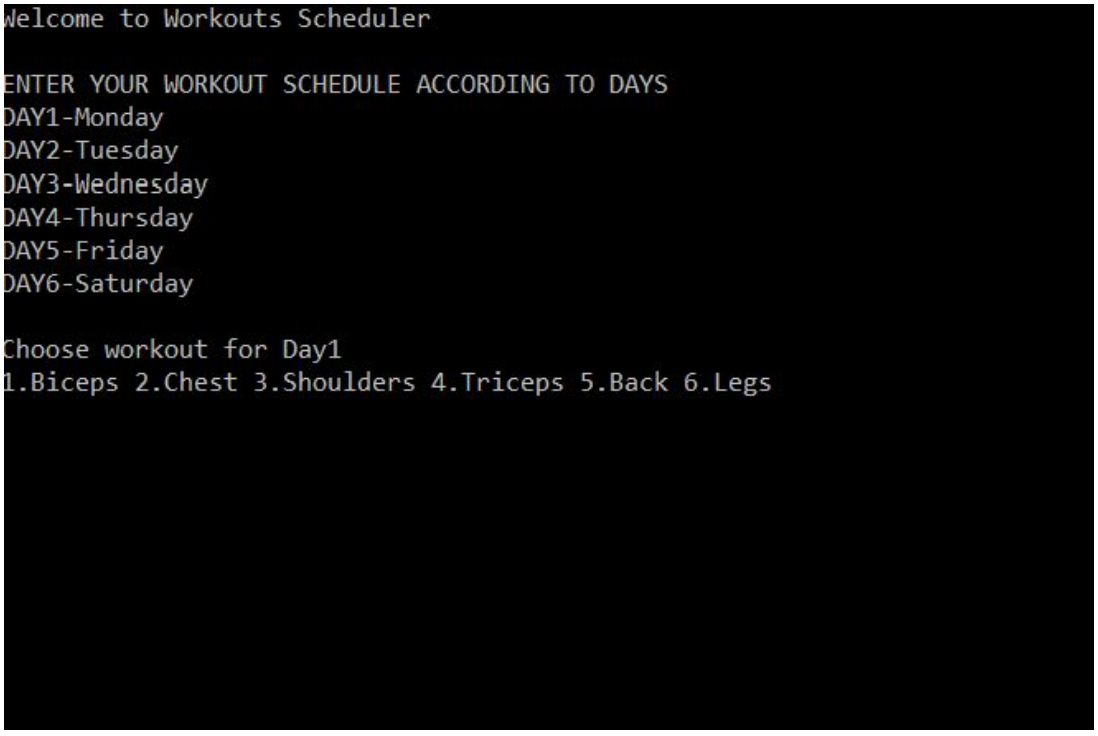
```
   { ob.setup(i,x);
         cout<<"Workout:"<<ob.workout[i]<<"\tEnter Next"<<endl;
         }
   else
   {
    cout<<"Wrong Input, Ener AGAIN!!";
    goto w;
   }
  }
 system("cls");
 cout<<"Workouts Initialized Succesfully"<<endl;
 getch();
 system("cls");
 while(l)
 {
  display(ob);
  l=ob.submenu();
}
}
```

The main function provides the sequential creation of classes and execution of all the functions in an organized and userfriendly manner.

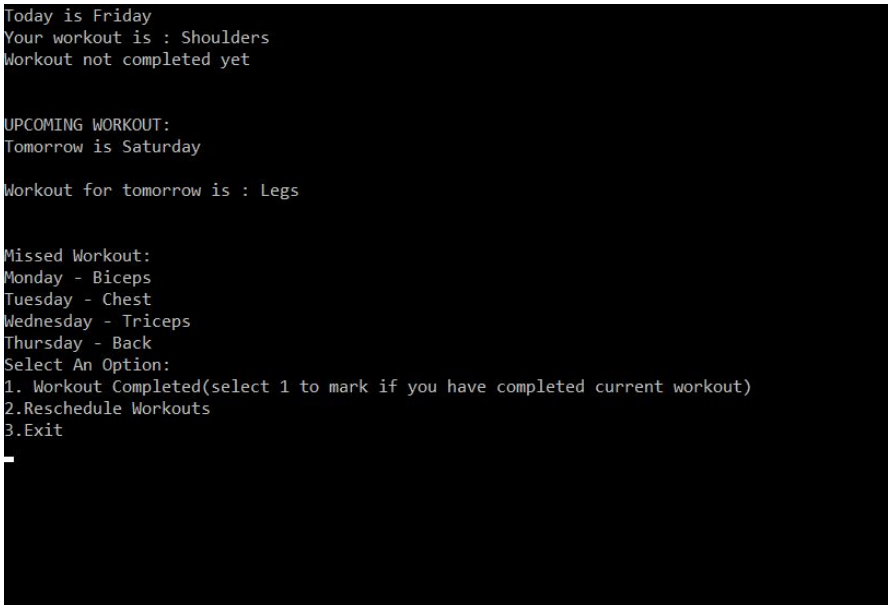
CHAPTER 5

OUTPUT SNAPSHOT

1)First the workouts needs to be scheduled according to the user's routine.



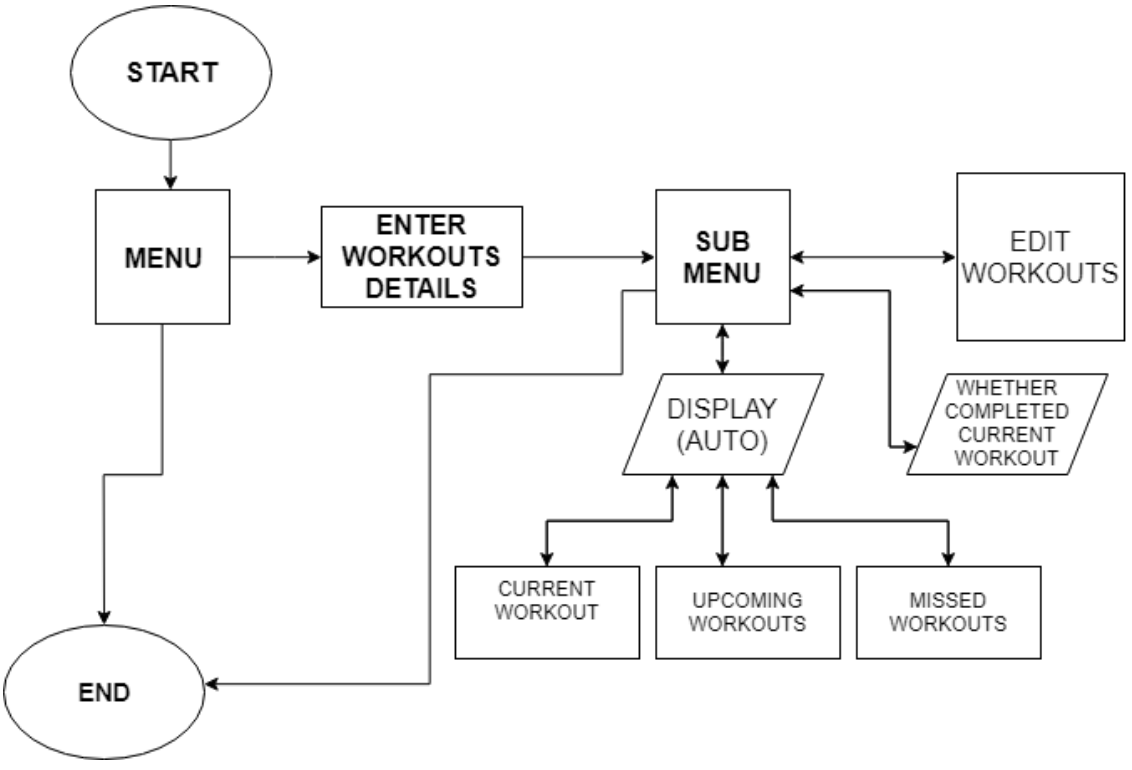
2)After Scheduling the workouts, current workout, missed workouts and upcoming workouts are displayed to the user.



3)Workout routine can be changed and edited by selecting the day and the workout.

```
Displaying Current Schedule :
DAY:            WORKOUT:
1.Monday                Biceps
2.Tuesday               Chest
3.Wednesday             Shoulders
4.Thursday              Triceps
5.Friday                Back
6.Saturday              Legs
Choose which day to be changed(1-6) *0 to cancel
3
Choose Workout to replace previous one
1.Biceps 2.Chest 3.Shoulders 4.Triceps 5.Back 6.Legs
3
Successfully updated
```

Program flow chart



CHAPTER 6

CONCLUSION

This project has taught me a wider way to implement and work on object oriented programming and exceeds the usage of its concepts as its limited to our lab programs, It has given me a whole new insight on how the idea of using objects as a way to solve our problems.

This Project has provided me a great insight on problem solving and debugging because of its complex yet simple code and solutions.

There were many hurdles and limitations, But the joy of getting the output without any errors makes it feel worth it.

Finally I would like to conclude by saying that this project has helped me gain a wider knowledge and understanding on how object oriented programming concepts makes it simpler to design and develop projects in an easier and much efficient way.

CHAPTER 7

REFERENCES

1. Wikibooks.org
2. C++-Programming Techniques (PCD) – A.M.Padma Reddy
3. https://Stackoverflow.com/linked_lists#
4. https://Quora.in
5. https://www.hackerrank.com/domains/OOPSconcepts