DR. B.R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY, JALANDHAR (144011), PUNJAB

OBJECT ORIENTED PROGRAMING PROJECT

CSPC-223

Number Mania

SUBMITTED BY:

KESHAV GOYAL (20103075)

KHUSHI (20103079)

MANDEEP SAINI (20103087)

SUBMITTED TO: DR. NONITA MAM

INTRODUCTION

Number Mania is a small mathematical game which helps participants to encourage them building their concepts of mathematical operations like addition, subtraction, multiplication.

FEATURES

The game opens with the screen displaying the name of the game then it ask the participant to enter the name

The game is divided into 4 parts

- ADDITION
- SUBTRACTION
- MULTIPLICATION
- CHALLENGE

Among these participant can choose any one of the above operations. Further all the operations are divided into 3 levels in order of their difficulty name level 1, level 2 and level 3 respectively

Each level consists of 5 question numbered 1 -5 .The system displays a positive message like “well done” is the answer entered by the participant is correct, on the other side it displays incorrect for every wrong answer.

After answering all the 5 questions the system calculates the accuracy of the participant in the level. The participant is allowed to move to next level only if the accuracy is more than 75% otherwise the participant is told to consult the teacher for topic clearance

At the end of the game the system displays the result of participant which consists score, accuracy, number of question answered and the comment based on the performance of the participant

OOPS CONCEPT

❖Encapsulation

The wrapping up of data and function into a single unit is known as encapsulation

Here the class named data is created which gets the information form the participants about the game

And functions named getdata , result ,calculate are declared in the class itself

```
class data{
    char name[50]; //name of the player
    //function variables: void calculation();
    int obt;
    double accuracy;

    void calculate(); //calculates the result
    void result();

    public:
    void getdata(); //takes input from the player
    void showdata() const; //prints data on screen
    void show_tabular() const;
};
```

❖Creation of objects

Objects are the basic run time entities in the object oriented programing. Objects are created by the help of class and from one class multiple objects can be created. Here we have created an object named player for the class data

Which represents one user at a time of execution of game

```
}
void write_player(){//writes data in file
    data player;
    ofstream outFile;
    outFile.open("numberMania.dat",ios::binary|ios::app);
    player.getdata();
    outFile.write(reinterpret_cast<char *> (&player),sizeof(data));
    outFile.close();

    cin.get();
}

void display_all()
{
    data player;
    ifstream inFile;
    inFile.open("numberMania.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        cin.ignore();
        cin.get();
        return;
    }
```

❖ Abstraction

It refers to the act of representation of essential without including the background information. It refers to the declaration of data into public and private

In the class name data is divided as following

Private:

Following functions and variables are declared as private

1. Char name
2. int obt
3. double accuracy
4. calculate()
5. result()

Public:

Following functions and variables are declared as public

1. getdata()
2. showdata()
3. show_tabular()

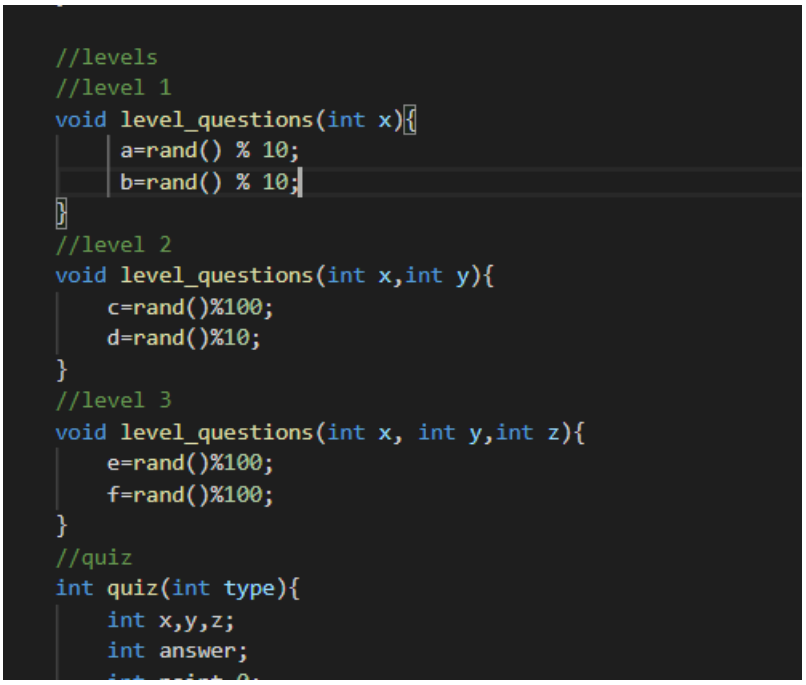
```
class data{
    char name[50]; //name of the player
    //function variables: void calculation();
    int obt;
    double accuracy;

    void calculate(); //calculates the result
    void result();

    public:
    void getdata(); //takes input from the player
    void showdata() const; //prints data on screen
    void show_tabular() const;
};
```

❖ Polymorphism

The type of polymorphism used in the project is function overloading. A function is said to be overloaded when we have more than one function with the same name but different parameter types or a different number of arguments.

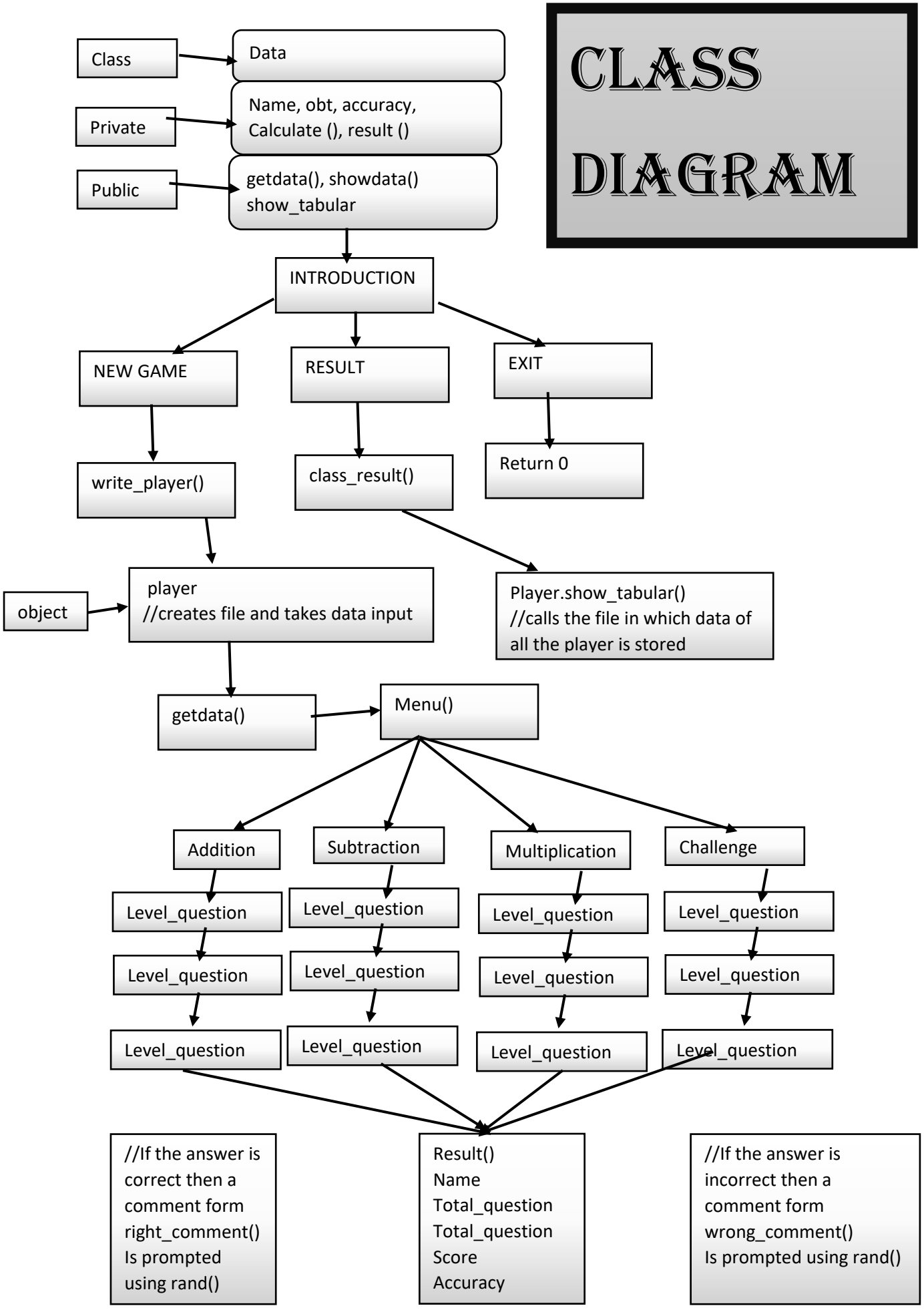
Thus a function can be overloaded based on the parameter types, the order of parameters and the number of parameters.

Here a function named level_question is defined three times but with different parameters which makes the code simple ,efficient and easy to understand



```
//levels
//level 1
void level_questions(int x){
    a=rand() % 10;
    b=rand() % 10;
}
//level 2
void level_questions(int x,int y){
    c=rand()%100;
    d=rand()%10;
}
//level 3
void level_questions(int x, int y,int z){
    e=rand()%100;
    f=rand()%100;
}
//quiz
int quiz(int type){
    int x,y,z;
    int answer;
```

CLASS DIAGRAM



CODE

```
//                          NUMBER MANIA
// >> The game consists of three levels based on the difficulty of questions.
// >> Each level consists of 5 questions based on the topic selected by the
player.
// >> The player can reach next level if and only if all the answers of the
previous levels are correct.

#include <iostream>
#include <cstring>
#include <cctype>
#include <stdio.h>
#include <iostream>
#include <conio.h>
#include <cstdio>
#include <ctime>
#include <Windows.h>
#include <fstream>
#include <iomanip>
#include <time.h>

using namespace std;

int myarr=0;
string mydata[50][5];
int score=0;
int que=0;
char ch;
int a,b,c,d,e,f;
string comm;
void gotoXY(int x, int y);
// void timer();
void write_player(); //stores player data
void introduction(); //first screen
void menu(); //addition,subtraction,multiplication,challenge
void level_questions(int a);
void level_questions(int a,int b);
void level_questions(int a,int b,int c);
void quiz(int type); //set the questions here
void display_all(); //reads all records from the file
void class_result();//tabular format
void result(); //display result
void right_comment(); //comments
void wrong_comment();
enum color{
    NONE,
```

```
    DARK_BLUE,
    GREEN,
    DARK_CYAN,
    DARK_RED,
    PURPLE,
    DARK_YELLOW,
    NORMAL,
    GRAY,
    BLUE,
    LIME,
    CYAN,
    RED,
    PINK,
    YELLOW,
    WHITE
};
void setcolor(color newColor);

int main(){
    char name;
    cout.setf(ios::fixed|ios::showpoint);
    cout<<setprecision(2);
    introduction();

    _getch();
    system("cls");

    return 0;
}

class data{
    char name[50]; //name of the player
    //function variables: void calculation();
    int obt;
    double accuracy;
    void calculate(); //calculates the result
    void result();

    public:
    void getdata(); //takes input from the player
    void showdata() const; //prints data on screen
    void show_tabular() const;
};

void data::calculate(){
    accuracy=score*100/que;

        if(accuracy>=80)
        {
```

```
            comm="Excellent";
            }
        else if(accuracy>=60)
            {
                comm="Good";
            }

        else if(accuracy>=40)
            {
                comm="Fair";
            }
        else
            {
                comm="Work Hard..";
            }
}

void data::getdata(){
    score=0;
    setcolor(BLUE);
    gotoXY(55,15);
    cout<<"Enter the Name of Player ==> ";
    cin.ignore();
    cin.getline(name,50);
    system("cls");
    menu();
    obt=score;
    calculate();
    system("cls");
    result();
}

void data::showdata() const{

    cout<<"\n Name of Player: "<<name;
}

void data::show_tabular() const{
    cout<<setw(20)<<name<<setw(10)<<obt<<setw(10)<<accuracy<<setw(6)<<endl;
    myarr++;
}


void gotoXY(int x, int y)   //function to decide location of the screem
{
    HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD CursorPosition;
    CursorPosition.X = x; // Locates column
    CursorPosition.Y = y; // Locates Row
```

```
    SetConsoleCursorPosition(console,CursorPosition); // Sets position for next
thing to be printed
}

void data::result()
{
    int f=30;
    Beep(2000, 100);
    Beep(1500, 100);
    Beep(2000, 100);
    Beep(1500, 100);
    Beep(2000, 700);
    Beep(0, 400);
    system("CLS");
        system("COLOR 5f");
    setcolor(RED);
    gotoXY(0,3);
    cout<<"\t\t\t\t\t\t  _____                 _ _   "<<endl;
    gotoXY(0,4);
    cout<<"\t\t\t\t\t\t |  __ \\               | | |  "<<endl;
    gotoXY(0,5);
    cout<<"\t\t\t\t\t\t | |__) |___  ___ _   _| | |_ "<<endl;
    gotoXY(0,6);
    cout<<"\t\t\t\t\t\t |  _  // _ \\/ __| | | | | __|"<<endl;
    gotoXY(0,7);
    cout<<"\t\t\t\t\t\t | | \\ \\  __/\\__ \\ |_| | | |_ "<<endl;
    gotoXY(0,8);
    cout<<"\t\t\t\t\t\t |_|  \\_\\___||___/\\__,_|_|\\__|"<<endl;

    setcolor(YELLOW);
gotoXY(f,11);
cout<<".-----------------------------------------------------------------."<<endl;
gotoXY(f,12);
cout<<"| .-------------------------------------------------------------. |"<<endl;
gotoXY(f,13);
cout<<"| |                                                             | |"<<endl;
gotoXY(f,14);
cout<<"| |                                                             | |"<<endl;
gotoXY(f,15);
cout<<"| |                                                             | |"<<endl;
gotoXY(f,16);
cout<<"| |                                                             | |"<<endl;
gotoXY(f,17);
cout<<"| |                                                             | |"<<endl;
gotoXY(f,18);
cout<<"| |_____________________________________________________________| |"<<endl;
gotoXY(f,19);
cout<<"|_________________________________________________________________|"<<endl;
    setcolor(CYAN);
```

```
    gotoXY(35,14);
    cout<<"Name ==>"<<name;
    setcolor(CYAN);
    gotoXY(35,15);
    cout<<"Total Score ==> "<< que*10;
    gotoXY(70,15);
    cout<<"Your Score ==> "<<score*10;
    setcolor(CYAN);
    gotoXY(35,16);
    cout<<"Accuracy ==> "<<accuracy;
    setcolor(CYAN);
    gotoXY(70,16);
    cout<<"Comments==> "<<comm;
_getch();

}
void setcolor   (color newColor)
{
    SetConsoleTextAttribute( GetStdHandle(STD_OUTPUT_HANDLE), (newColor ) );
}

void right_comment(){
    srand(time(0));
 switch(rand()%10)
 {
     case 0:
     {
         cout<<"Perfect Answer";
         break;
     }
     case 1:
     {
         cout<<"Ideal  Answer";
         break;
     }
     case 2:
     {
         cout<<"correct  Answer";
         break;
     }
     case 3:
     {
         cout<<"Excellent  Answer";
         break;
     }
     default:
     {
         cout<<"you got that correct ";
     }
```

```
 }

}

void wrong_comment()
{
    srand(time(0));
 switch(rand()%10)
 {
     case 0:
     {
         cout<<"better luck next time ";
         break;
     }
     case 1:
     {
         cout<<"Ask your teacher for help";
         break;
     }
     case 2:
     {
         cout<<"incorrect answer ";
         break;
     }
     case 3:
     {
         cout<<"not good";
         break;
     }
     default:
     {
         cout<<"So sorry,its a wrong answer ";
     }
 }

}

//levels
//level 1
void level_questions(int x){
     a=rand() % 10;
     b=rand() % 10;
}
//level 2
void level_questions(int x,int y){
    c=rand()%100;
    d=rand()%10;
}
//level 3
```

```
void level_questions(int x, int y,int z){
    e=rand()%100;
    f=rand()%100;
}
//quiz
void quiz(int type){
    int x,y,z;
    int answer;
    int point=0;
    switch(type){
        case 1: //addition
        {
            gotoXY(45,4);
            cout<<"Level 1"<<endl;
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x);
                    system("CLS");
                    system("COLOR 4f");

            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<a<<" + "<<b<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==a+b){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                point++;
                score++;
                getch();

            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }

        que+=5;
        if(point>3){
            point=0;
            Beep(0,4000);
```

```
            gotoXY(56,3);
            system("CLS");
            gotoXY(50,5);
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 2"<<endl;
            getch();
                for(int i=1;i<=5;i++){
                level_questions(x,y);
                    system("CLS");
                    system("COLOR 5f");

            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<c<<" + "<<d<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==c+d){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                point++;
                score++;
                getch();
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }
        if(point>3){
            point=0;
            system("CLS");
            gotoXY(50,5);
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 3"<<endl;
            getch();
```

```
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x,y,z);
                    system("CLS");
                    system("COLOR 6f");

            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<e<<" + "<<f<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==e+f){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }

    }
    break;
    case 2:
        {
            gotoXY(45,4);
            cout<<"Level 1"<<endl;
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x);
                    system("CLS");
                    system("COLOR 4f");
            setcolor(BLUE);
            gotoXY(23,5);
            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<a<<" - "<<b<<endl;
            gotoXY(45,15);
```

```
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==a-b){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        if(point>3){
            point=0;
            system("CLS");
            Beep(0,4000);
            gotoXY(50,5);
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 2"<<endl;
            getch();
                for(int i=1;i<=5;i++){
                    level_questions(x,y);
                    system("CLS");
                    system("COLOR 5f");
            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<c<<" - "<<d<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==c-d){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
```

```
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }
        if(point>3){
            point=0;
            system("CLS");
            gotoXY(50,5);
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 3"<<endl;
            getch();
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x,y,z);
                    system("CLS");
                    system("COLOR 6f");

            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<e<<" - "<<f<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==e-f){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }
```

```
    }
    break;
    case 3:
        {
            gotoXY(45,4);
            cout<<"Level 1"<<endl;
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x);
                    system("CLS");
                    system("COLOR 4f");
            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<a<<" * "<<b<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==a*b){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        if(point>3){
            point=0;
            Beep(0,4000);
            system("CLS");
            gotoXY(50,5);
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 2"<<endl;
            getch();
                for(int i=1;i<=5;i++){
                    level_questions(x,y);
                    system("CLS");
```

```
                system("COLOR 5f");
            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<c<<" * "<<d<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==c*d){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }
        if(point>3){
            point=0;
            system("CLS");
            gotoXY(50,5);
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 3"<<endl;
            getch();
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x,y,z);
                    system("CLS");
                    system("COLOR 6f");
            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<e<<" * "<<f<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
```

```
            gotoXY(62,15);
            cin>>answer;
            if(answer==f*e){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }
    }
    break;
    case 4:
        {
            gotoXY(45,4);
            cout<<"Level 1"<<endl;
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x);
                    system("CLS");
                    system("COLOR 4f");
            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<a<<" + "<<b<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==a+b){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
```

```
            }
        }
        que+=5;
        if(point>3){
            point=0;
            system("CLS");
            Beep(0,4000);
            gotoXY(50,5);
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 2"<<endl;
            getch();
                for(int i=1;i<=5;i++){
                    level_questions(x);
                    level_questions(x,y);
                    level_questions(x,y,z);
                    system("CLS");
                    system("COLOR 5f");
            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<c<<" + "<<d<<" - "<<e<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==c+d-e){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }
        if(point>3){
            point=0;
            system("CLS");
            gotoXY(50,5);
```

```
            cout<<"NEXT LEVEL"<<endl;
            gotoXY(50,6);
            cout<<"Loading..."<<endl;
            gotoXY(45,4);
            cout<<"Level 3"<<endl;
            getch();
            Beep(0,4000);
                for(int i=1;i<=5;i++){
                    level_questions(x);
                    level_questions(x,y);
                    level_questions(x,y,z);
                    system("CLS");
                    system("COLOR 6f");
            setcolor(BLUE);
            gotoXY(23,5);

            cout<<"Question "<<i<<": "<<endl;
            gotoXY(43,10);
            cout<<"( "<<a<<" * "<<b<<" ) "<<" + "<<c<<" - "<<d<<endl;
            gotoXY(45,15);
            cout<<"Your answer ==> ";
            gotoXY(62,15);
            cin>>answer;
            if(answer==(a*b)+c-d){
                gotoXY(55,16);
                right_comment();
                gotoXY(62,19);
                getch();
                point++;
                score++;
            }else{
                gotoXY(62,18);
                wrong_comment();
                getch();
            }
        }
        que+=5;
        }
    }
    break;
    default:
        cout<<"/n INVALID CHOICE";
        _getch();
    }
}
void write_player(){//writes data in file
    data player;
    ofstream outFile;
    outFile.open("numberMania.dat",ios::binary|ios::app);
```

```
    player.getdata();
    outFile.write(reinterpret_cast<char *> (&player),sizeof(data));
    outFile.close();

    cin.get();
}

void display_all()
{
    data player;
    ifstream inFile;
    inFile.open("numberMania.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        cin.ignore();
        cin.get();
        return;
    }
    cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
    while(inFile.read(reinterpret_cast<char *> (&player), sizeof(data)))
    {
        player.showdata();
        cout<<"\n\n====================================\n";
    }
    inFile.close();
    cin.ignore();
    cin.get();
}

void class_result()
{
    data player;
    ifstream inFile;

    inFile.open("numberMania.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        cin.ignore();

        cin.get();
        return;
    }

    cout<<"\n\n\t\tALL PLAYER RESULT \n";
    cout<<"\n\t\tPress Enter to Go Back Home Screen\n";
    cout<<"============================================================\n";
    cout<<"         Name      Points     Accuracy      "<<endl;
```

```
    cout<<"============================================================\n";
    while(inFile.read(reinterpret_cast<char *> (&player), sizeof(data)))
    {
        player.show_tabular();
    }
    cin.ignore();
    cin.get();
    inFile.close();
}
//first screen
void introduction(){

    doo :
    system("CLS");
    Beep(2000, 100);
    Beep(1500, 100);
    Beep(2000, 100);
    Beep(1500, 100);
    Beep(2000, 700);
    Beep(0, 400);
setcolor(LIME);
cout<<"\n\n\t\t\t
___    __                    __                          ___      ___"<<endl;
cout<<"\t\t\t|   \\  |  |                  |  |                       |   \\
/   |                  __"<<endl;
cout<<"\t\t\t|    \\ |  | __   __  _ __ ___  |  |___    ___   __
__    |    \\  /    |  ___ __  _ ___   __   ___ __"<<endl;
cout<<"\t\t\t|  \\  \\|  ||  | |  || '_ ` _ \\ |  /  _ \\  / __ \\|  '\\
/   |     \\/     | /   `| || '_  \\ |  | /   `| |"<<endl;
cout<<"\t\t\t|  |\\  \\  ||  \\_/  || | | | | ||    |_| ||   __/|
|       |  |\\    /|  ||  |  | || | |  ||  ||  |  | |"<<endl;
cout<<"\t\t\t|__| \\____| \\_____/ |_| |_| |_||__\\____/  \\____/|_|       |__|
\\__/ |__| \\___/|_||_| |__||__| \\___/|_|"<<endl;
    setcolor(GREEN);
    gotoXY(50,20);
    cout<<" >>";
    gotoXY(74,20);
    cout<<" >>";
    gotoXY(54,20);
    setcolor(WHITE);
    cout<<"Press Enter Continue";
    Beep(500, 100);
    gotoXY(45,10);
    cout<<"====== Select Following Option ======";
    gotoXY(50,12);
    cout<<"1-->  NEW GAME";
    gotoXY(50,13);
    cout<<"2-->  RESULT";
    gotoXY(50,14);
```

```
    cout<<"3-->  EXIT";
    gotoXY(42,15);
    cout<<"Select : ";
    Beep(500, 100);
    go:
    int type;
    gotoXY(51,15);
    cin>>type;
    switch(type)
    {
        case 1:
            {
                        system("CLS");

            system("COlOR 5f");
            write_player();
            break;
            }

        case 2:
            {
            system("CLS");
                system("COLOR 4f");
            class_result();

            system("CLS");
            goto doo;
            break;
            }

        case 3:
        {
            system("CLS");
            exit(0);
            break;
        }
         default:

            cout<<"/n INVALID CHOICE";
            _getch();
            goto go;
    }
    //system("CLS");
}
//second screen
void menu()
{

    doo :
```

```
    Beep(2000, 100);
    Beep(1500, 100);
    Beep(2000, 100);
    Beep(1500, 100);
    Beep(2000, 700);
    Beep(0, 400);
    system("CLS");
    setcolor(LIME);
cout<<"\n\n\t\t\t\t\t   ___       ___"<<endl;
cout<<"\t\t\t\t\t  |   \\    /   |"<<endl;
cout<<"\t\t\t\t\t  |    \\  /    |  ___    _ ___   __   __"<<endl;
cout<<"\t\t\t\t\t  |     \\/     | / __ \\ | '_  \\ |  | |  |"<<endl;
cout<<"\t\t\t\t\t  |  |\\    /|  ||   __/ | | |  ||  \\_/  |"<<endl;
cout<<"\t\t\t\t\t  |__| \\__/ |__| \\____/ |_| |__| \\_____/"<<endl;
setcolor(DARK_BLUE);
    gotoXY(50,20);
    cout<<" >>";
    gotoXY(74,20);
    cout<<" >>";
    gotoXY(54,20);
    setcolor(WHITE);
    cout<<"Press Enter Continue";
    Beep(500, 100);
    gotoXY(45,10);
    cout<<"======Select Following Option ======";
    gotoXY(50,12);
    cout<<"1-->  ADDITION";
    gotoXY(50,13);
    cout<<"2-->  SUBTRACTION";
    gotoXY(50,14);
    cout<<"3-->  MULTIPLICATION";
    gotoXY(50,15);
    cout<<"4-->  CHALLENGE";
    gotoXY(42,16);
    cout<<"Select : ";
    Beep(500, 100);
    go:
    int type;
    gotoXY(51,16);
    cin>>type;
    switch(type)
    {
        case 1:
            {
            system("CLS");
            setcolor(BLUE);
            quiz(1);
            break;
            }
```

```
        case 2:
            {
            system("CLS");
            setcolor(GREEN);
            quiz(2);
            break;
            }
        case 3:
            {
            system("CLS");
            setcolor(BLUE);
            quiz(3);
            break;
            }
        case 4:
            {
            system("CLS");
            setcolor(RED);
            quiz(4);
            break;
            }
         default:

            cout<<"/n INVALID CHOICE";
            _getch();
            goto go;
    }
    //system("CLS");
}
```