

**Dr BR Ambedkar National Institute of Technology
Jalandhar-144011, Punjab, India.**

July-Dec, 2021



Department of Computer Science & Engineering

**Object Oriented Programming Lab
(CSCP-233)**

PROJECT REPORT

GUESS THE NUMBER GAME

Submitted to:

Dr. Nonita Sharma

(Assistant Professor)

Department of Computer Science & Engineering

Submitted by:

Abhinandan Gautam -20103002

Aditya Kumar -20103006

Aditya Nagre -20103007

Amritdeep Singh-20103013

GROUP1(SECTIONA)

CSE II YEAR

INTRODUCTION:

Here in the project we are playing a game called 'Guess the number'. The program is set so that it automatically takes a random number that is to be guessed by the player in the range of 1 to 1000.

The player then tries to guess that number. As the player enters his/her guess, the program tells whether the guess is correct or not. If it aint a correct guess, it'll acknowledge the player about precision of his/her guess. The program will run into loop until the player guesses the correct number and telling about its precision every single time.

The program is modified to count the number of guesses the player makes. By taking 10 guesses as a borderline, the program will print a particular message based on the count.

If the count is less than 10 or exact 10 or greater than 10 the messages shown are "Either you know the secret or you got lucky", "haha you know the secret", "you should be able to do better why should it take no more than 10 guesses" respectively. With each good guess half of the numbers will be eliminated and the program can also show the secret of guessing the number in 10 steps.

FEATURES:

The program contains a class 'secret' and some other classes which have this class as a base class. This way the repetition of certain functions is avoided.

Some functions such as system("color x"), Sleep(x), setw, etc are used so as to make the user experience bit more attractive and interactive. Functions such as rand(x) is used to take a random shot on a number between 1 to 1000.

For easier interpretation the code is provided with descriptive cout statements.

Constructors and Destructors are used to initialise values to data members and give information about the end of the scope respectively.

Headers such as 'windows.h' and 'dos.h' are included to use certain functions directly, which makes the code less complex.

Each and every function is segregated on the basis of its functionality with name which broadly describes its functionality. So a different coder would also understand or make changes in the code with ease.

For handling complexity, making the code maintainable, ensuring its reusability some OOPS concepts are implemented in the program.

Basic functions used in the program are as follows:

❖ **set_no_of_guess**

```
Game :: set_no_of_guess(0);  
static void set_no_of_guess(int x){ no_of_guess=0;}  
here we are calling a static member function with parameter 0.
```

❖ **begin**

```
begin(G);  
void begin(Game &g){  
    int num = rand()%1000 + 1;  
  
    g.actual_num = num;  
}
```

Storing a random number in variable 'actual_num' of class Game.

❖ **guess_num**

```
G.guess_num();  
void guess_num()  
{  
    int guessNum;  
    system("color E");
```

```

    cout<<" Enter the guessed number : ";
    cin>>guessNum;
    if(guessNum>0 && guessNum<=1000)
    {
        no_of_guess++;
        guessed_num = guessNum;
    }
    else
    { system("color 4");
        cout<<" !!!! Wrong Input !!!!\n";
        cout<<" You select the number out of range.\n Select number between 1 to
1000\n";
        Sleep(1000);
        (*this).guess_num();
    }
}

```

guess_num() function is used here to input guessed number

if input number between 1 to 1000, then guessed_num = input number

else re-run function for that object

❖ check

```

    G.check();

void Game :: check(){
    if(guessed_num==actual_num)
    { system("color 2");
        flag = false;
        cout<<"\n\t\t\t -----";
        cout<<"\n\t\t\t Congratulations ^^ ^^ \n \t\t\t You guessed the number
correctly.\n";
        cout<<"\t\t\t -----\n\n";
        Sleep(1000);
        (*this).message();
    }
    else{

```

```

        (*this).help();
    }
}

```

check() function to check guessed number is low, high or correct

if guessed number = actual number run message function

else run help function

defining the function outside the class

❖ **friend begin**

```
friend void begin(Game &);
```

Declaration of friend function begin()

It begins the game and initializes the actual_num to a random number.

❖ **help**

```
void help()
```

```

{
    if(guessed_num<actual_num)
    { system("color 4");
      cout<<"\tOh, Wrong Guess\n";
      cout<<"\tActual Number is greater than guessed
number.\n"<<setw(22)<<"Retry\n\n";
      Sleep(1000);
      (*this).guess_num();
    }
    else
    { system("color 4");
      cout<<"\tOh, Wrong Guess\n";
      cout<<"\tActual Number is lower than guessed
number.\n"<<setw(22)<<"Retry\n\n";
      Sleep(1000);
      (*this).guess_num();
    }
}
}

```

help() function to check either number is low or high

❖ **message**

```

void message(){
    if(no_of_guess < 10)
    { system("color 2");
      cout<<" Number of Guess = "<<no_of_guess<<endl;
      cout<<" Either you know the secret or you got lucky\n\n";
    }
    else if(no_of_guess == 10)
    { system("color 2");
      cout<<" Number of Guess = "<<no_of_guess<<endl;
      cout<<" Ha Ha Ha... You know the secret\n\n";
    }
    else
    { system("color B");
      cout<<" Number of Guess = "<<no_of_guess<<endl;
      cout<<" You should be able to do better.\n Why should it take no more than
10 guesses\n\n";
      cout<<"Are you excited to Know the secret of finding number in less than 10
guesses : ";
      cout<<"YES\n";
      cout<<setw(84)<<" NO\n\n";
      cout<<"Enter the choice : ";
      string s;
      cin>>s;
      if(s=="YES" || s=="Yes" || s=="yes"){
          explain_the_secret(actual_num);//is-a relation (Class game has a object of
class secret)//class inherited
      }}}

```

message() function print the accurate message.

message is displayed according to number of guesses.

There are certain if-elseif-else conditions according to the no. of guesses by the player.

❖ **explain_the_secret**

```

void explain_the_secret(int num){
    system("color B");

```

```

Sleep(1200);

cout<<"\nThe secret is : HALF-INTERVAL SEARCH\n\n";Sleep(3000);

cout<<setw(84)<<"          -----          \n";
cout<<setw(84)<<"          LeTs TRy It          \n";
cout<<setw(84)<<"          -----          \n\n";

cout<<"First choose end and start point of your range i.e 0 and 1000\n";
int a=0;int b=1000;
int x;
while(1){
    x=(a+b)/2;

    cout<<"\tGuess the middle number of your given range i.e : "<<x<<"\n";
    cout<<"          -----          \n";

    Sleep(5000);

    if(x<num){
        a=(a+b)/2; system("color B");

        cout<<"\nYour guess is smaller than actual number.";Sleep(750);
        cout<<"\n\tAll the number below this number are not your answer
\n";Sleep(750);

        cout<<"\t\tNow, Range become "<<a<<" to "<<b<<endl;Sleep(750);}

    else if(x>num){
        b=(a+b)/2; system("color B");

        cout<<"\nYour guess is greater than actual number.";Sleep(750);
        cout<<"\n\tAll the number above this number are not your answer
\n";Sleep(750);

        cout<<"\t\tNow, Range become "<<a<<" to "<<b<<endl;Sleep(750);}

    else{
        break;}}

    cout<<"\nIn this way you can guess the number in less than 10 guesses\n";}

```

explain_the_secret is a function to explain how one can guess the actual/correct no. in less than 10 guesses.

The process here is similar to the binary search in arrays.

After every guess we are shortening our range and then taking its mid value and comparing it with the guessed no.

OOPS CONCEPTS:

INHERITANCE:

It is the procedure in which one class inherits the attributes and methods of another class. In place of writing the same code, again and again, we can simply inherit the properties of one class into the other.

```
class Parent
{
    public:
        int id_p;
};
class Child : public Parent
{
    public:
        int id_c;
};
```

In our 'guess the number' program we have taken the class 'secret' as a parent/base class and then a class 'game' is defined as its child class with 'public' access mode. In this way the member function of the 'secret' class such as explain_the_secret is accessible to 'game' class.

```
class secret{
public:
    void explain_the_secret(int num);
};
class Game : public secret{}
```

- is-a relation: IS-A is a totally based on Inheritance, which can be of two types Class Inheritance or Interface Inheritance.

"A is a B type of thing".

Wherever you see an extends keyword or implements keyword in a class declaration, then this class is said to have IS-A relationship.

In our program, class 'game' has a object of class 'secret'.

ENCAPSULATION:

Encapsulation is defined as wrapping up of data and information under a single unit. In Object Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulates them.

In our 'guess the number' program we have taken a class named 'secret' and a member function is defined in this class, which can only be accessed by an object of class 'secret'.

```
class secret{
public: void explain_the_secret(int num);
```

In the same way in class 'game' two data members and some member functions are bonded together. This is nothing but Encapsulation.

```
class Game : public secret
{
private:
```

```

    int actual_num;
    int guessed_num;
    static int no_of_guess;
public:
    static void set_no_of_guess(int x);
    void guess_num();
    void help();void message();void check();

```

ABSTRACTION:

Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

In our 'guess the number' program Abstraction is one of the basic OOPS concept that is used. We have taken classes like 'secret' and 'game' in which the data members or variables are not allowed to access directly rather we need to call certain member functions to access them.

CLASS/OBJECT:

A class in C++ is the building block, that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.

An Object is an instance of a Class.

In our 'guess the number' program classes such as 'secret' and 'game' are defined.

```
friend void begin(Game &);
```

An object 'G' is defined in our given program.

```
class secret{};
```

```
class Game : public secret{};
```

CONCEPTS RELATED TO CLASS/OBJECT:-

- The concept of friend function(friend function can be given a special grant to access private and protected members) is also related to classes and objects.
Game G;
- Constructor and Destructor are also associated with classes/objects.
Constructor for class 'game':-
Game(int num1=0, int num2=0)
{
 actual_num = num1;
 guessed_num = num2;
};
- Destructor for class 'game':-
~Game(){};
- This pointer contains the address of the invoking object.
(*this).guess_num();
- Static Member Function:- By declaring a function member as static, you make it independent of any particular object of the class. A static member function can be called even if no objects of the class exist and the static functions are accessed using only the class name and the scope resolution operator.
static void set_no_of_guess(int x);

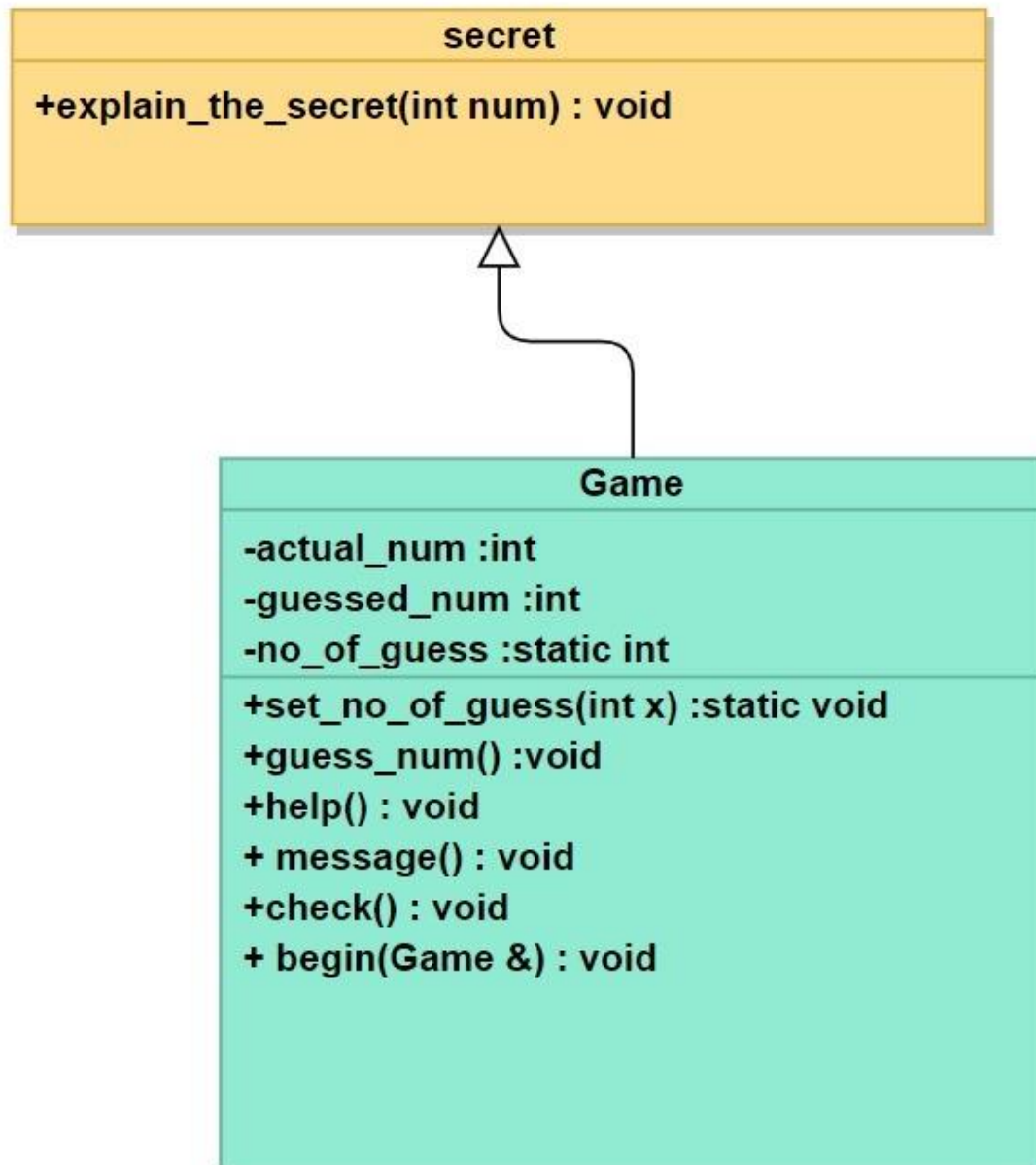
.

SOME OTHER CONCEPTS:

- **sleep Function:** The sleep () function causes the program or the process in which it is called, to suspend its execution temporarily for a period of time in seconds specified by the function parameter. Execution is suspended until the requested time(in milliseconds) is elapsed or a signal or an interrupt is delivered to the function.
- **setw Function:** It is used to sets the field width to be used on output operations. This function is basically in the 'iomanip' header.
- **system("color colorcode"):** system() is included in Windows.h library. Colorcodes as follows:-

0 = Black	8 = Gray
1 = Blue	9 = Light Blue
2 = Green	A = Light Green
3 = Aqua	B = Light Aqua
4 = Red	C = Light Red
5 = Purple	D = Light Purple
6 = Yellow	E = Light Yellow
7 = White	F = Bright White
- **rand Function:** rand() function is to get random number.
- **srand Function:** The srand() function sets the starting point for producing a series of pseudo-random integers. If srand() is not called, the rand() seed is set as if srand(1) were called at program start. Any other value for seed sets the generator to a different starting point.

CLASS DIAGRAM:



CODE:

```
// Guess the number game
#include <bits/stdc++.h>
#include <windows.h>
#include <dos.h>
#include <ctime>
using namespace std;
bool flag = true;
class secret
{
public:
    void explain_the_secret(int num)
    {
        system("color B");
        Sleep(1200);
        cout << "\nThe secret is : HALF-INTERVAL SEARCH\n\n";
        Sleep(3000);
        cout << setw(84) << "      -----          \n";
        cout << setw(84) << "          LeTs TRy It          \n";
        cout << setw(84) << "      -----          \n\n";
        cout << "First choose end and start point of your range i.e 0 and 1000\n";
        int a = 0;
        int b = 1000;
        int x;
        while (1)
        {
            x = (a + b) / 2;
            cout << "\tGuess the middle number of your given range i.e : " << x << "\n";
            cout << "      -----          \n";
            Sleep(5000);
            if (x < num)
```

```

{
    a = (a + b) / 2;
    system("color B");
    cout << "\nYour guess is smaller than actual number.";
    Sleep(750);
    cout << "\n\tAll the number below this number are not your answer \n";
    Sleep(750);
    cout << "\t\tNow, Range become " << a << " to " << b << endl;
    Sleep(750);
}
else if (x > num)
{
    b = (a + b) / 2;
    system("color B");
    cout << "\nYour guess is greater than actual number.";
    Sleep(750);
    cout << "\n\tAll the number above this number are not your answer \n";
    Sleep(750);
    cout << "\t\tNow, Range become " << a << " to " << b << endl;
    Sleep(750);
}
else
{
    break;
}
}
cout << "\nIn this way you can guess the number in less than 10 guesses\n";
}
};

//Inheritance of secret class
class Game : public secret
{

```

private:

```
int actual_num;  
int guessed_num;  
static int no_of_guess; /*Static member*/
```

public:

```
//Static member function  
static void set_no_of_guess(int x)  
{  
    no_of_guess = 0;  
}
```

/*

-> constructor

-> default arguments

-> constructor overloading

*/

```
Game(int num1 = 0, int num2 = 0)
```

```
{  
    actual_num = num1;  
    guessed_num = num2;  
}
```

/*

Declaration of friend function begin()

*/

```
friend void begin(Game &);
```

/*

-> guess_num() function to input guessed number

-> if input number between 1 to 1000, then guessed_num = input number

-> else re-run function for that object

-> this pointer

*/

```
void guess_num()
```

```

{
    int guessNum;
    system("color E");
    cout << " Enter the guessed number : ";
    cin >> guessNum;

    if (guessNum > 0 && guessNum <= 1000)
    {
        no_of_guess++;
        guessed_num = guessNum;
    }
    else
    {
        system("color 4");
        cout << " !!!! Wrong Input !!!!\n";
        cout << " You select the number out of range.\n Select number between 1 to 1000\n";
        Sleep(1000);
        (*this).guess_num();
    }
}

/*
-> help() function to check either number is low and high
*/

void help()
{
    if (guessed_num < actual_num)
    {
        system("color 4");
        cout << "\tOh, Wrong Guess\n";
        cout << "\tActual Number is greater than guessed number.\n"
            << setw(22) << "Retry\n\n";
        Sleep(1000);
    }
}

```



```

        (*this).guess_num();
    }
else
{
    system("color 4");
    cout << "\tOh, Wrong Guess\n";
    cout << "\tActual Number is lower than guessed number.\n"
        << setw(22) << "Retry\n\n";
    Sleep(1000);
    (*this).guess_num();
}
}
/*

```

-> message() function print the accurate message

-> message is displayed according to number of guesses

*/

```

void message()
{
    if (no_of_guess < 10)
    {
        system("color 2");
        cout << " Number of Guess = " << no_of_guess << endl;
        cout << " Either you know the secret or you got lucky\n\n";
    }
    else if (no_of_guess == 10)
    {
        system("color 2");
        cout << " Number of Guess = " << no_of_guess << endl;
        cout << " Ha Ha Ha... You know the secret\n\n";
    }
    else
    {

```

```

        system("color B");
        cout << " Number of Guess = " << no_of_guess << endl;
        cout << " You should be able to do better.\n Why should it take more than 10
guesses\n\n";
        cout << "Are you excited to Know the secret of finding number in less than 10
guesses : ";
        cout << "YES\n";
        cout << setw(84) << " NO\n\n";
        cout << "Enter the choice : ";
        string s;
        cin >> s;
        if (s == "YES" || s == "Yes" || s == "yes")
        {
            explain_the_secret(actual_num); //is-a relation (Class game has a object of class
secret)//class inherited
        }
    }
}
/*

```

-> check() function to check guessed number is low, high or correct

-> this pointer

-> if guessed number = actual number run message function

-> else run help function

-> defining the function out the class

*/

```
void check();
```

```
~Game() {} //Destructor
```

```
};
```

```
/* ->Friend function of class Game
```

->Used to initialize the actual_number

-> rand() function to get random number

-> rand()%1000 + 1 to get number between 1 to 1000

*/

```

void begin(Game &g)
{
    int num = rand() % 1000 + 1;

    g.actual_num = num;
}

/* This function is defined outside the class using scope resolution operator :: */
void Game ::check()
{
    if (guessed_num == actual_num)
    {
        system("color 2");
        flag = false;
        cout << "\n\t\t\t -----";
        cout << "\n\t\t\t Congratulations ^^ ^^ \n\t\t\t You guessed the number correctly.\n";
        cout << "\t\t\t ----- \n\n";
        Sleep(1000);
        (*this).message();
    }
    else
    {
        (*this).help();
    }
}

int Game ::no_of_guess = 0;

int main()
{
    srand(time(NULL));
    system("color 3");
    cout << setw(84) << " ----- \n";
    cout << setw(84) << "          GUESS THE NUMBER          \n";
    cout << setw(84) << " ----- \n\n";
}

```

```

cout << setw(72) << " Welcome to Guess The Number Game, \n\n";
string name;
cout << "Enter your name : ";
cin >> name;
cout << "\n\tHello " << name << "\n\n";
while (1)
{
    Game ::set_no_of_guess(0); //Call to static member function
    flag = true;
    system("color c");
    int choice;
    Sleep(1200);
    cout << "\n *Press 1 to enter * \n";
    cout << " *Press 0 to exit * \n";
    Sleep(1000);
    cout << "\n Enter choice : ";
    cin >> choice;
    if (choice == 1)
    {
        cout << "\t\t\tAll THE BeSt\n\n";
        Game G;
        cout << " Lets Begin ..... \n\n";
        begin(G);
        cout << " I have a number between 1 to 1000. Can you guess my number?\n\n";
        G.guess_num();
        while (flag)
        {
            G.check();
        }
    }
    else if (choice == 0)
    {

```

```
        break;
    }
    else
    {
        cout << " Press Correct Choice \n";
    }
    Sleep(2000);
    cout << setw(84) << "      -----ThAnX FoR PlAyInG-----
\n\n";
    system("Pause");
}
return 0;
}
```

OUTPUT:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ASUS> cd "c:\Users\ASUS\Desktop\C++\OOPS_LAB_2\" ; if ($?) { g++ 2.cpp -o 2 } ; if ($?) { .\2 }

-----
                        GUESS THE NUMBER
-----

                        Welcome to Guess The Number Game,

Enter your name : aditya

                        Hello aditya

*Press 1 to enter *
*Press 0 to exit *

Enter choice : 1

                        All The BeSt

Lets Begin .....

I have a number between 1 to 1000. Can you guess my number?

Enter the guessed number : 500
Oh, Wrong Guess
Actual Number is greater than guessed number.
Retry

Enter the guessed number : 750
Oh, Wrong Guess
Actual Number is greater than guessed number.
Retry
```

```
Enter the guessed number : 875
Oh, Wrong Guess
Actual Number is lower than guessed number.
Retry

Enter the guessed number : 813
Oh, Wrong Guess
Actual Number is lower than guessed number.
Retry

Enter the guessed number : 766
Oh, Wrong Guess
Actual Number is lower than guessed number.
Retry

Enter the guessed number : 758
Oh, Wrong Guess
Actual Number is greater than guessed number.
Retry

Enter the guessed number : 762
Oh, Wrong Guess
Actual Number is lower than guessed number.
Retry

Enter the guessed number : 760
```

```
-----
Congratulations ^^ ^^
You guessed the number correctly.
-----
```

```
Number of Guess = 9
Either you know the secret or you got lucky
```

```
-----ThAnX FoR pLAyInG-----
```

Press any key to continue . . .

*Press 1 to enter *
*Press 0 to exit *

Enter choice : 1

All The BeSt

Lets Begin

I have a number between 1 to 1000. Can you guess my number?

Enter the guessed number : 1

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 2

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 3

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 4

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 4

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 5

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 6

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 45

Oh, Wrong Guess

Actual Number is greater than guessed number.

Retry

Enter the guessed number : 500

Oh, Wrong Guess

Actual Number is lower than guessed number.

Retry

Enter the guessed number : 400

Oh, Wrong Guess

Actual Number is lower than guessed number.

Retry

Enter the guessed number : 300

Oh, Wrong Guess

```

        Actual Number is lower than guessed number.
        Retry

Enter the guessed number : 100
Oh, Wrong Guess
        Actual Number is lower than guessed number.
        Retry

Enter the guessed number : 50
Oh, Wrong Guess
        Actual Number is greater than guessed number.
        Retry

Enter the guessed number : 60
Oh, Wrong Guess
        Actual Number is greater than guessed number.
        Retry

Enter the guessed number : 80
Oh, Wrong Guess
        Actual Number is greater than guessed number.
        Retry

Enter the guessed number : 101
Oh, Wrong Guess
        Actual Number is lower than guessed number.
        Retry

Enter the guessed number : 81
Oh, Wrong Guess
        Actual Number is greater than guessed number.
        Retry

```

```

Enter the guessed number : 44
Oh, Wrong Guess
        Actual Number is greater than guessed number.
        Retry
Enter the guessed number : 1500
!!!! Wrong Input !!!!
You select the number out of range.
Select number between 1 to 1000
Enter the guessed number : 90

-----
Congratulations ^^ ^^
You guessed the number correctly.
-----

Number of Guess = 34
You should be able to do better.
Why should it take no more than 10 guesses

Are you excited to Know the secret of finding number in less than 10 guesses : YES
NO

Enter the choice : YES

The secret is : HALF-INTERVAL SEARCH

-----
LeTs TRy It
-----

First choose end and start point of your range i.e 0 and 1000

```



```

Number of Guess = 34
You should be able to do better.
Why should it take no more than 10 guesses

Are you excited to Know the secret of finding number in less than 10 guesses : YES
NO

Enter the choice : YES

The secret is : HALF-INTERVAL SEARCH

-----
LeTs TRy It
-----

First choose end and start point of your range i.e 0 and 1000
Guess the middle number of your given range i.e : 500
-----

Your guess is greater than actual number.
All the number above this number are not your answer
Now, Range become 0 to 500
Guess the middle number of your given range i.e : 250
-----

Your guess is greater than actual number.
All the number above this number are not your answer
Now, Range become 0 to 250
Guess the middle number of your given range i.e : 125
-----

```

```

Your guess is greater than actual number.
All the number above this number are not your answer
Now, Range become 0 to 125
Guess the middle number of your given range i.e : 62
-----

Your guess is smaller than actual number.
All the number below this number are not your answer
Now, Range become 62 to 125
Guess the middle number of your given range i.e : 93
-----

Your guess is greater than actual number.
All the number above this number are not your answer

Your guess is smaller than actual number.
All the number below this number are not your answer
Now, Range become 89 to 93
Guess the middle number of your given range i.e : 91
-----

Your guess is greater than actual number.
All the number above this number are not your answer
Now, Range become 89 to 91
Guess the middle number of your given range i.e : 90
-----

In this way you can guess the number in less than 10 guesses
-----ThAnX FoR PLAyInG-----

Press any key to continue . . . █

```

GitHub Repository Link

<https://github.com/AbhinandanGautam/OOPS-Project>