

INDEX

S.No	Description	Page No
1	AIM	1
2	OVERIEW OF C++	2
3	SYNOPSIS	3
4	REQUIREMENTS	4
5	HEADER FILES	6
6	FUNCTIONS	7
7	ALGORITHM	8
8	SOURCE CODE	9
9	OUTPUT	26
10	CONCLUSION	32
11	BIBLIOGRAPHY	33

AIM

To use C++ to create a sudoku puzzle.

OVERVIEW

C++ is a statically typed, compiled, general purpose, case sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.

It is regarded as a middle-level programming language as it comprises a combination of both high-level and low-level language features.

C++ was developed by Bjarne Stroustrup in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the existing C language. It was originally named C with Classes but was later renamed to C++ in 1983. It is a superset of C, and virtually, any legal C program is a legal C++ program.

SYNOPSIS

We have used C++ to create a Sudoku game in which the player is given three lives.

REQUIREMENTS

(HARDWARE AND SOFTWARE)

WINDOWS

HARDWARE	
RECOMMENDED	Intel Core™ 2 Duo processor or Intel Xeon processor or higher
MINIMUM REQUIREMENTS	One of the following: <ul style="list-style-type: none"> • Intel Pentium 4 processor family and higher • Intel® Xeon Phi™ coprocessor • Non-Intel processors compatible with above processors

SOFTWARE

RECOMMENDED	<ul style="list-style-type: none"> ○ Intel Parallel Studio XE 2013 SPI ○ Intel Parallel Studio XE 2015 Professional Edition
MINIMUM REQUIREMENTS	<ul style="list-style-type: none"> • One of the following OS: <ul style="list-style-type: none"> ○ Microsoft Windows 8, 8.1 ○ Microsoft Windows 7 SP 1 ○ Microsoft Windows Server 2012 ○ Microsoft Windows 2008 ○ Microsoft Windows HPC Server • One of the following compilers: <ul style="list-style-type: none"> ○ Intel C++ Compiler 13.1 and higher ○ Microsoft Visual Compiler 2010 and higher

LINUX	
HARDWARE	
RECOMMENDED	<ul style="list-style-type: none"> • Intel Core 2 Duo processor or higher • Intel Xeon Phi coprocessor
MINIMUM REQUIREMENTS	<ul style="list-style-type: none"> • One of the following: <ul style="list-style-type: none"> ○ Intel Pentium 4 processor family ○ Intel® Xeon coprocessor or higher ○ Non-Intel but corresponding processors
SOFTWARE	
MINIMUM REQUIREMENTS	<ul style="list-style-type: none"> • One of the following OS: <ul style="list-style-type: none"> ○ Red Hat Enterprise Linux 5, 6, 7 ○ Fedora 20 ○ Debian 6.0, 7 ○ Intel Cluster Ready • One of the following compilers: <ul style="list-style-type: none"> ○ Intel C++ Compiler 13.1 ○ Parallel Studio XE 2013
RECOMMENDED	<ul style="list-style-type: none"> • Intel Parallel Studio XE 2013 SP1 • Intel Parallel Studio XE 2015 Professional Studio

HEADER FILES

The following header files were included in the program:

- `iostream.h`: To include `cin`, `cout`.
- `conio.h`: To include `clrscr()`, `getch()`.
- `process.h`: To include `exit()`.

FUNCTIONS

The following built-in functions are used in the program:

- clrscr(): Function to clear the output.
- getch(): Function to hold the output window until hitting any key from the keyboard.
- exit(): Function to exit the program.

A function was defined to display the Sudoku puzzle after each entry of the answer.

```
void display(char s[][25])
{
    cout<<"\n-----";
    for(int i=0;i<6;i++)
    {
        cout<<"\n";
        for(int j=0;j<25;j++)
            cout<<s[i][j];
        if(i==1||i==3)
            cout<<"\n-----";
    }
    cout<<"\n-----";
    return;
}
```


ALGORITHM

Step 1: Start

Step 2: Create a Sudoku puzzle using a 2D character array

Step 3: $\text{life} \leftarrow 3$, $\text{count} \leftarrow 24$.

Step 4: Print the puzzle.

Step 5: Enter position and corresponding answer

Step 6: If the answer is correct then $\text{count} = \text{count} - 1$. Print the puzzle with the answer in place of the position entered.

Step 7: Else, $\text{life} \leftarrow \text{life} - 1$, $\text{count} \leftarrow \text{count} - 1$.

Step 8: If the user enters 0, exit program.

Step 9: Else if $\text{life} = 0$ or $\text{count} = 0$, exit program.

Step 10: Else, go back to step 5.

Step 11: If $\text{count} = 0$, print "Congratulations!"

Step 12: Else print "Better luck next time!"

Step 13: Stop.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
void display(char s[][25])
{
    cout<<"\n-----";
    for(int i=0;i<6;i++)
    {
        cout<<"\n";
        for(int j=0;j<25;j++)
            cout<<s[i][j];
        if(i==1||i==3)
            cout<<"\n-----";
    }
    cout<<"\n-----"<<endl;
    return;
}
void main()
{
```

```

clrscr();

char s[6][25]={ "| 4 A B | C D 3 |", "| E 3 1 | F G H |",
               "| I J K | 3 6 L |", "| M 2 N | 5 O P |",
               "| 1 Q R | S T 5 |", "| U V 3 | W 2 X |"};

char pos, ans;

int count=24, life=3;

char sol[24]={'6', '5', '2', '1', '2', '4', '5', '6', '5', '1', '4', '2', '3',
             '6', '4', '1', '4', '2', '6', '3', '6', '5', '1', '4'};

do
{
    clrscr();

    cout<<"\n SUDOKU:";

    cout<<"\n RULES:\n 1. Positions must be entered in uppercase only!";

    cout<<"\n 2. You have three lives which will decrease with a wrong answer.";

    cout<<"\n 3. Press enter to continue.";

    cout<<"\n 4. Enter 0 to quit.";

    cout<<"\n 5. ENJOY!";

    display(s);

```

```
cout<<"Enter position: ";
cin>>pos;
if (pos=='0')
{
    exit(0);
}
cout<<"Enter number: ";
cin>>ans;
switch(pos)
{
    case 'A': if(ans==sol[0])
        {
            s[0][6]=ans;
            count--;
        }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
}
```

```

        break;
case 'B': if(ans==sol[1])
    {
        s[0][9]=ans;
        count--;
    }
else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'C': if(ans==sol[2])
    {
        s[0][15]=ans;
        count--;
    }
else
    {
        cout<<"Incorrect!";

```

```

        life--;
    }
    break;
case 'D': if(ans==sol[3])
    {
        s[0][18]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'E': if(ans==sol[4])
    {
        s[1][3]=ans;
        count--;
    }
    Else

```

```

        {
            cout<<"Incorrect!";
            life--;
        }
        break;
case 'F': if(ans==sol[5])
    {
        s[1][15]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'G': if(ans==sol[6])
    {
        s[1][18]=ans;
        count--;
    }

```

```

    }
else
{
    cout<<"Incorrect!";
    life--;
}
break;
case 'H': if(ans==sol[7])
{
    s[1][21]=ans;
    count--;
}
else
{
    cout<<"Incorrect!";
    life--;
}
break;
case 'T': if(ans==sol[8])
{

```



```

        s[2][3]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'J': if(ans==sol[9])
    {
        s[2][6]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;

```

```

case 'K': if(ans==sol[10])
    {
        s[2][9]=ans;
        count--;
    }
else
    {
        cout<<"Incorrect!";
        life--;
    }
break;
case 'L': if(ans==sol[11])
    {
        s[2][21]=ans;
        count--;
    }
else
    {
        cout<<"Incorrect!";
        life--;
    }

```

```

    }
    break;
case 'M': if(ans==sol[12])
    {
        s[3][3]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'N': if(ans==sol[13])
    {
        s[3][9]=ans;
        count--;
    }
    else
    {

```

```

        cout<<"Incorrect!";
        life--;
    }
    break;
case 'O': if(ans==sol[14])
    {
        s[3][18]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'P': if(ans==sol[15])
    {
        s[3][21]=ans;
        count--;
    }

```

```

else
{
    cout<<"Incorrect!";
    life--;
}
break;
case 'Q': if(ans==sol[16])
{
    s[4][6]=ans;
    count--;
}
else
{
    cout<<"Incorrect!";
    life--;
}
break;
case 'R': if(ans==sol[17])
{
    s[4][9]=ans;

```

```

        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'S': if(ans==sol[18])
    {
        s[4][15]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'T': if(ans==sol[19])

```

```

    {
        s[4][18]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'U': if(ans==sol[20])
    {
        s[5][3]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }

```

```

        break;
case 'V': if(ans==sol[21])
    {
        s[5][6]=ans;
        count--;
    }
else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;
case 'W': if(ans==sol[22])
    {
        s[5][15]=ans;
        count--;
    }
else
    {
        cout<<"Incorrect!";

```



```

        life--;
    }
    break;
case 'X': if(ans==sol[23])
    {
        s[5][21]=ans;
        count--;
    }
    else
    {
        cout<<"Incorrect!";
        life--;
    }
    break;

}

cout<<"\n Lives left: "<<life;
getch();
} while((count!=0)&&(life!=0));
if (count==0)

```

```
{  
    clrscr();  
    display(s);  
    cout<<"\nCONGRATULATIONS!";  
    cout<<"\nYou Did It!";  
}  
else  
    cout<<"\n Better luck next time ";  
    getch();  
}
```

OUTPUT

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 A B | C D 3 |  
| E 3 1 | F G H |  
-----
```

```
-----  
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
-----  
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: A

Enter number: 6

Lives left: 3

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 B | C D 3 |  
| E 3 1 | F G H |  
-----
```

```
-----  
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
-----  
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: B

Enter number: 2

Incorrect!

Lives left: 2_

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 B | C D 3 |  
| E 3 1 | F G H |  
-----
```

```
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: B

Enter number: 5

Lives left: 2_

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 5 | C D 3 |  
| E 3 1 | F G H |  
-----
```

```
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: E

Enter number: 2

Lives left: 2_

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 5 | C D 3 |  
| 2 3 1 | F G H |  
-----
```

```
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: C

Enter number: 2

Lives left: 2

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 5 | 2 D 3 |  
| 2 3 1 | F G H |  
-----
```

```
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: D

Enter number: 1

Lives left: 2

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 5 | 2 1 3 |  
| 2 3 1 | F G H |  
-----
```

```
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: G

Enter number: 5

Lives left: 2_

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 5 | 2 1 3 |  
| 2 3 1 | F 5 H |  
-----
```

```
-----  
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
-----  
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: F

Enter number: 4

Lives left: 2

SUDOKU:

RULES:

1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```
-----  
| 4 6 5 | 2 1 3 |  
| 2 3 1 | 4 5 H |  
-----
```

```
-----  
| I J K | 3 6 L |  
| M 2 N | 5 0 P |  
-----
```

```
-----  
| 1 Q R | S T 5 |  
| U V 3 | W 2 X |  
-----
```

Enter position: H

Enter number: 6

Lives left: 2

```

SUDOKU:
RULES:
1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```

```

-----
| 4 6 5 | 2 1 3 |
| 2 3 1 | 4 5 6 |
|-----|

```

```

| I J K | 3 6 L |
| M 2 N | 5 0 P |
|-----|

```

```

| 1 Q R | S T 5 |
| U V 3 | W 2 X |
|-----|

```

```

Enter position: L
Enter number: 1
Incorrect!
Lives left: 1_

```

```

SUDOKU:
RULES:
1. Positions must be entered in uppercase only!
2. You have three lives which will decrease with a wrong answer.
3. Press enter to continue.
4. Enter 0 to quit.
5. ENJOY!

```

```

-----
| 4 6 5 | 2 1 3 |
| 2 3 1 | 4 5 6 |
|-----|

```

```

| I J K | 3 6 L |
| M 2 N | 5 0 P |
|-----|

```

```

| 1 Q R | S T 5 |
| U V 3 | W 2 X |
|-----|

```

```

Enter position: 0_

```

The game is quitted, and the program ends.

CONCLUSION

In the completion of this project, we have used different concepts of C++. We used do while loop, for loop, switch case, if statements, built in function like exit (0) and user defined function.