



Display on **Console** at
a Specific Location

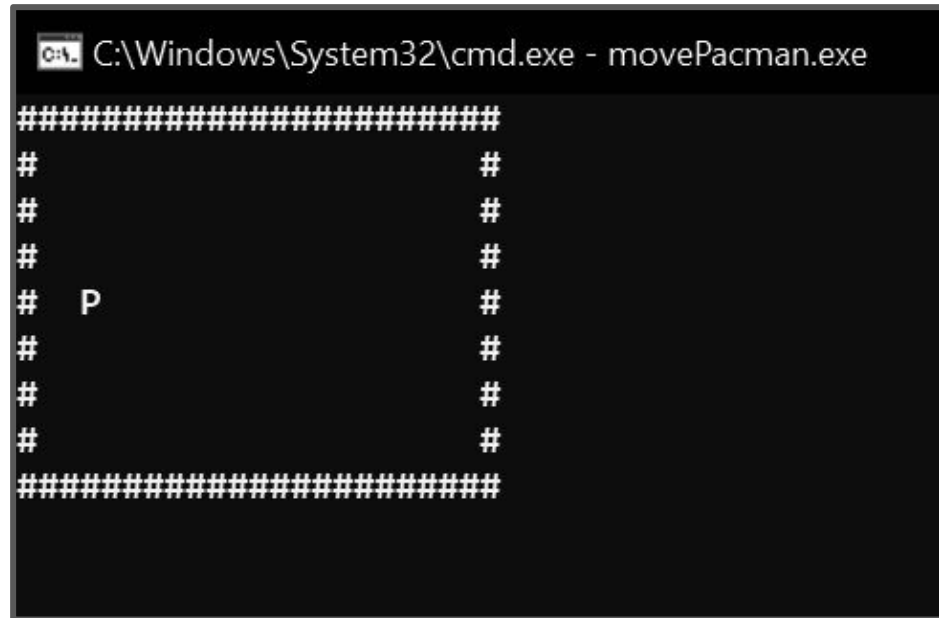


اَللّٰهُمَّ ارْزُقْنِيْ عِلْمًا نَّافِعًا وَاسِعًا عَمِيْقًا

اَللّٰهُمَّ ارْزُقْنِيْ رِزْقًا وَّاسِعًا حَلَالًا طَيِّبًا
مُّبَارَكًا مِنْ عِنْدِكَ

Goal: Display Pacman on the Console

The goal is to display the Pacman in the maze.



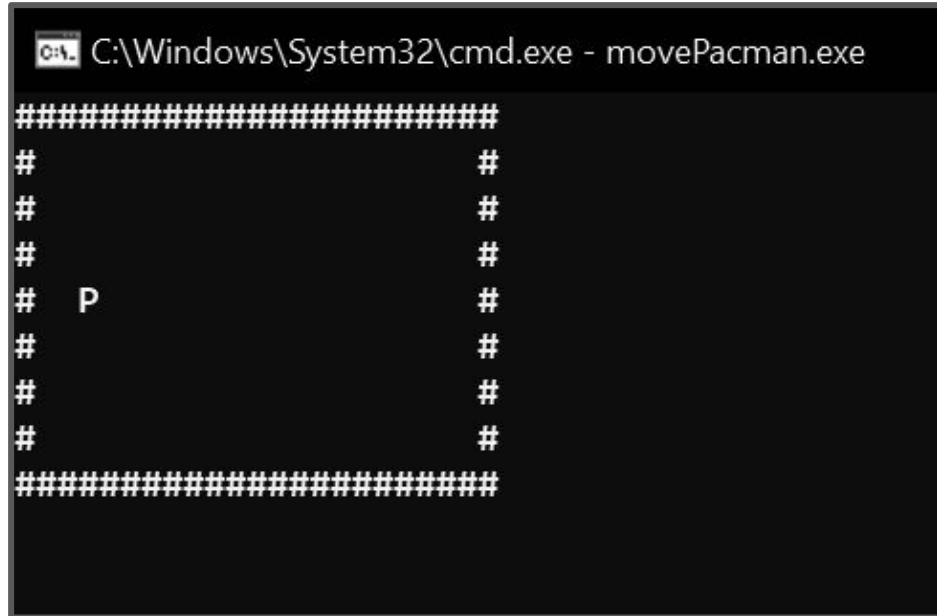
```
C:\Windows\System32\cmd.exe - movePacman.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
#                                     #
#                                     #
#                                     #
#####
```

A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\System32\cmd.exe - movePacman.exe". The window content displays a maze represented by a grid of '#' characters. The maze is 8 rows high and 20 columns wide. The top and bottom rows are solid lines of '#' characters. The second through seventh rows have '#' characters at the first and last positions, with the rest of the space being empty. In the fourth row, at the second column position, there is a white 'P' character representing Pacman.

Goal: Move Pacman on the Console

Lets see First, how we can place a character on a specific location on the console.



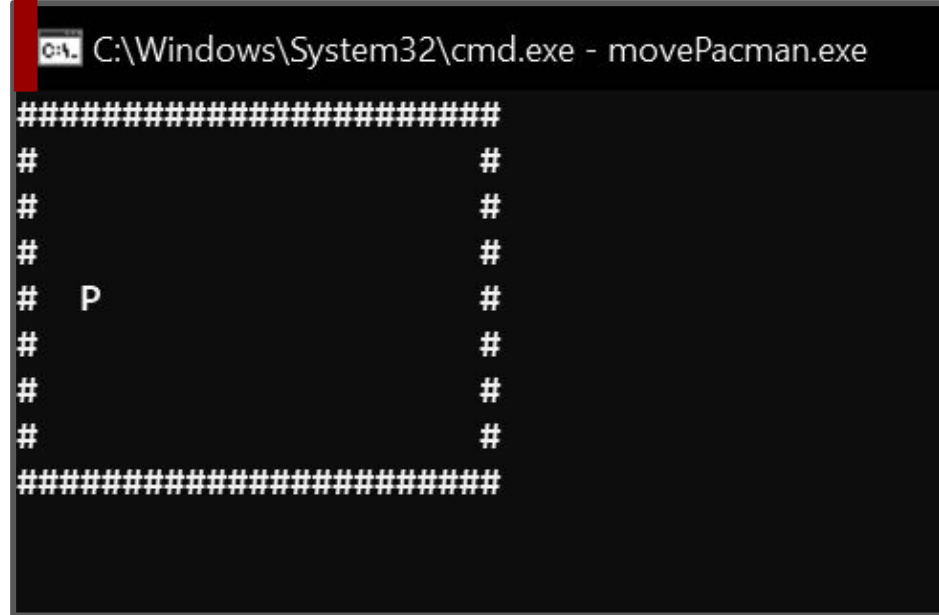
```
C:\Windows\System32\cmd.exe - movePacman.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
#                                     #
#                                     #
#                                     #
#####
```

The screenshot shows a Windows command prompt window with the title bar "C:\Windows\System32\cmd.exe - movePacman.exe". The window content displays a console box made of hash symbols (#). Inside the box, the letter 'P' is placed on the fourth line from the top, representing Pacman. The box is 8 lines high and 20 characters wide.

X and Y Coordinates of Console

(X,Y)
(0,0)

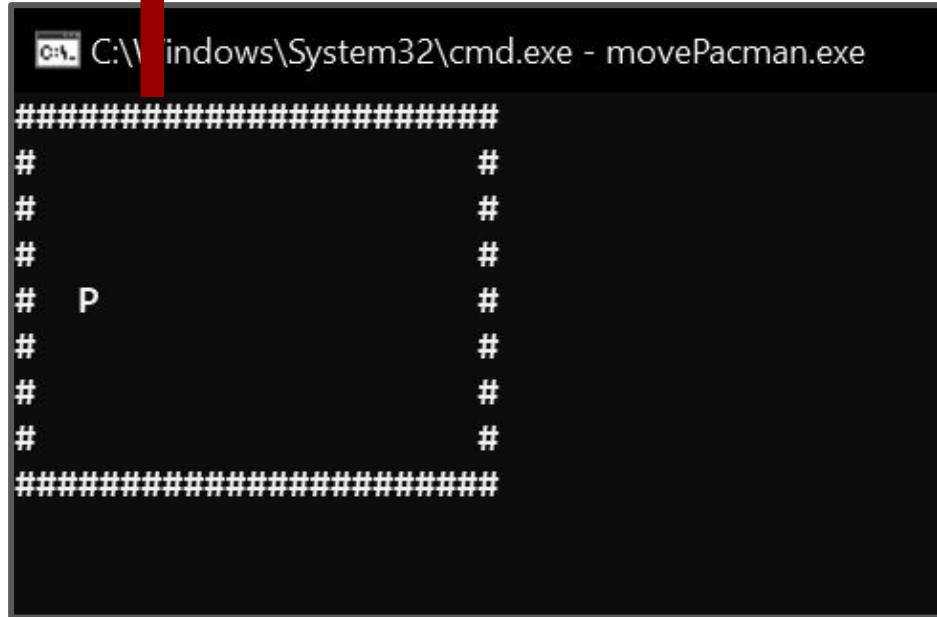


```
C:\Windows\System32\cmd.exe - movePacman.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
#                                     #
#                                     #
#                                     #
#####
```

X and Y Coordinates of Console

(6,0) (X,Y)



```
C:\Windows\System32\cmd.exe - movePacman.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
#                                     #
#                                     #
#                                     #
#####
```

X and Y Coordinates of Console

(X, Y)
(0, 4) ←

```
C:\Windows\System32\cmd.exe - movePacman.exe

#####
#                                     #
#                                     #
#                                     #
# P                                   #
#                                     #
#                                     #
#                                     #
#####
```


X and Y Coordinates of Console

(X,Y)
(3,4) ← P

```
C:\Windows\System32\cmd.exe - movePacman.exe

#####
#                                     #
#                                     #
#                                     #
# P                                  #
#                                     #
#                                     #
#                                     #
#####
```

Place the cursor on specific location

We will use a function `gotoxy()` and we will pass it the **x coordinates** of the console and **y coordinates** of the console and it will place the cursor on specific location on console.

```
#####  
#                                     #  
#                                     #  
#                                     #  
#  P                                 #  
#                                     #  
#                                     #  
#                                     #  
#####
```

|| gotoxy() Function

To use `gotoxy()` function we have to include `windows.h` file.

```
#include <windows.h>
```

gotoxy() Function

The definition of **gotoxy()** function is given by:

```
void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

gotoxy() Function

We will not go into the functionality of this function, we will just copy this function in our program and use it.

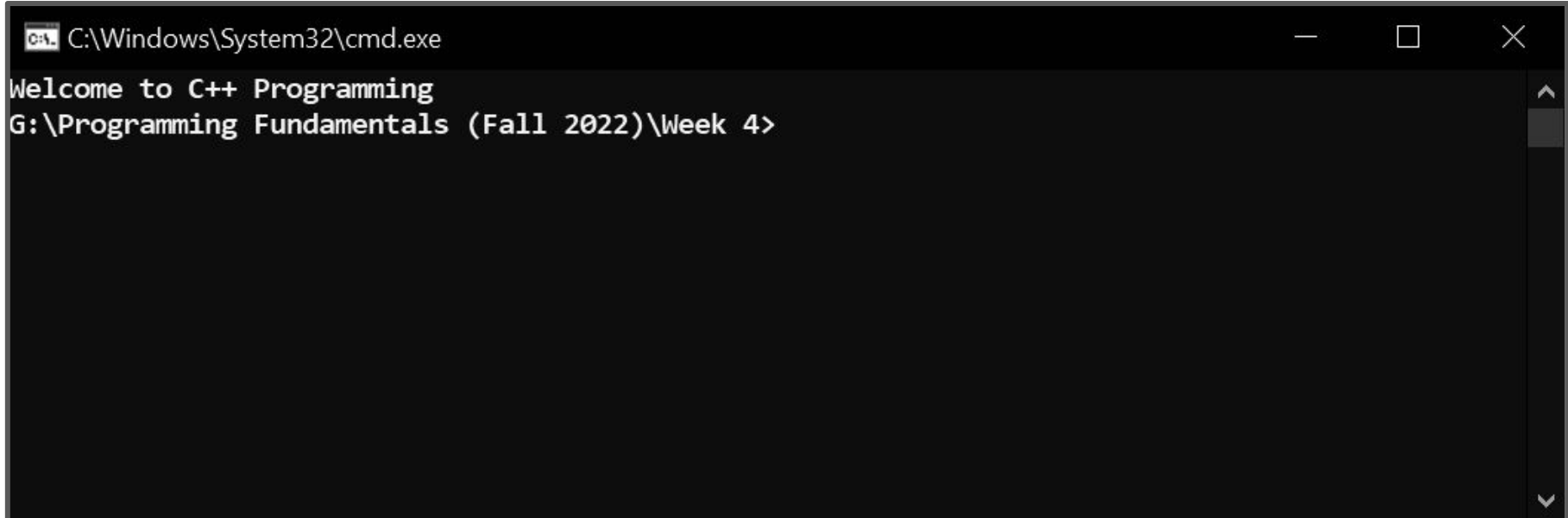
```
void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 01

Let's write "Welcome to C++ Programming" at the (0,0) coordinates of console

Activity 01

Let's write "Welcome to C++ Programming" at the (0,0) coordinates of console



```
C:\Windows\System32\cmd.exe
Welcome to C++ Programming
G:\Programming Fundamentals (Fall 2022)\Week 4>
```

A screenshot of a Windows Command Prompt window. The title bar shows the file name 'C:\Windows\System32\cmd.exe' and standard window controls (minimize, maximize, close). The command prompt area has a black background with white text. The first line of text is 'Welcome to C++ Programming' at the top left. The second line shows the current directory 'G:\Programming Fundamentals (Fall 2022)\Week 4' followed by a prompt character '>'. A vertical scrollbar is visible on the right side of the window.

Activity 01: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);
main()
{
    // Write your Code here
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

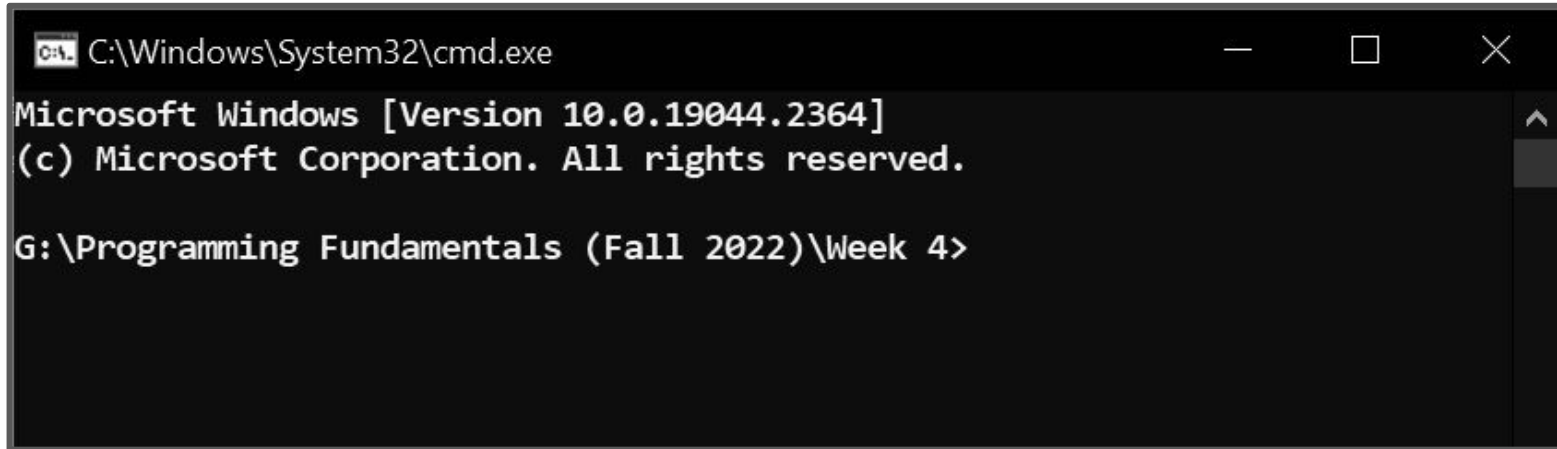

Activity 01: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);
main()
{
    gotoxy(0, 0);
    cout << "Welcome to C++ Programming";
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 01: Solution

When you just open the cmd it is displaying the Microsoft Windows Version at X = 0 coordinate.

A screenshot of a Windows Command Prompt window. The title bar shows the file path 'C:\Windows\System32\cmd.exe' and standard window controls (minimize, maximize, close). The command prompt displays the following text: 'Microsoft Windows [Version 10.0.19044.2364]' followed by '(c) Microsoft Corporation. All rights reserved.' on the next line. The current directory is 'G:\Programming Fundamentals (Fall 2022)\Week 4' and the prompt is 'G>'.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

G:\Programming Fundamentals (Fall 2022)\Week 4>
```

Activity 01: Solution

So, after compiling and executing the program, we will get the following output.



```
C:\Windows\System32\cmd.exe
```

```
Welcome to C++ Programming 10.0.19044.2364]
```

```
G:\Programming Fundamentals (Fall 2022)\Week 4>
```

```
G:\Programming Fundamentals (Fall 2022)\Week 4>c++ print.cpp -o print.exe
```

```
G:\Programming Fundamentals (Fall 2022)\Week 4>print.exe
```

Activity 01: Solution

Although, it has displayed “Welcome to C++ Programming” but it has override the previous data and the output is not making sense.



```
C:\Windows\System32\cmd.exe
```

```
Welcome to C++ Programming 10.0.19044.2364]
```

```
G:\Programming Fundamentals (Fall 2022)\Week 4>
```

```
G:\Programming Fundamentals (Fall 2022)\Week 4>c++ print.cpp -o print.exe
```

```
G:\Programming Fundamentals (Fall 2022)\Week 4>print.exe
```

Activity 01: Solution

So, what is the solution of this?

A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\Windows\System32\cmd.exe' and standard window controls (minimize, maximize, close). The command history is as follows:
1. 'Welcome to C++ Programming 10.0.19044.2364]'
2. 'G:\Programming Fundamentals (Fall 2022)\Week 4>'
3. 'G:\Programming Fundamentals (Fall 2022)\Week 4>c++ print.cpp -o print.exe'
4. 'G:\Programming Fundamentals (Fall 2022)\Week 4>print.exe'
The text is white on a black background. A vertical scrollbar is visible on the right side of the command area.

C:\Windows\System32\cmd.exe

Welcome to C++ Programming 10.0.19044.2364]

G:\Programming Fundamentals (Fall 2022)\Week 4>

G:\Programming Fundamentals (Fall 2022)\Week 4>c++ print.cpp -o print.exe

G:\Programming Fundamentals (Fall 2022)\Week 4>print.exe

Activity 01: Solution

We must clear the screen first.

A screenshot of a Windows Command Prompt window. The title bar shows the icon for Command Prompt and the path 'C:\Windows\System32\cmd.exe'. The window has standard Windows window controls (minimize, maximize, close) on the right. The command prompt shows a welcome message, the current directory, and two commands being executed.

C:\Windows\System32\cmd.exe

Welcome to C++ Programming 10.0.19044.2364]

G:\Programming Fundamentals (Fall 2022)\Week 4>

G:\Programming Fundamentals (Fall 2022)\Week 4>c++ print.cpp -o print.exe

G:\Programming Fundamentals (Fall 2022)\Week 4>print.exe

Activity 01: Solution

In order to call the `cls` command in C++, we have to use `system("cls")` function with input parameter `cls` as string.



C:\Windows\System32\cmd.exe

Welcome to C++ Programming 10.0.19044.2364]

G:\Programming Fundamentals (Fall 2022)\Week 4>

G:\Programming Fundamentals (Fall 2022)\Week 4>c++ print.cpp -o print.exe

G:\Programming Fundamentals (Fall 2022)\Week 4>print.exe

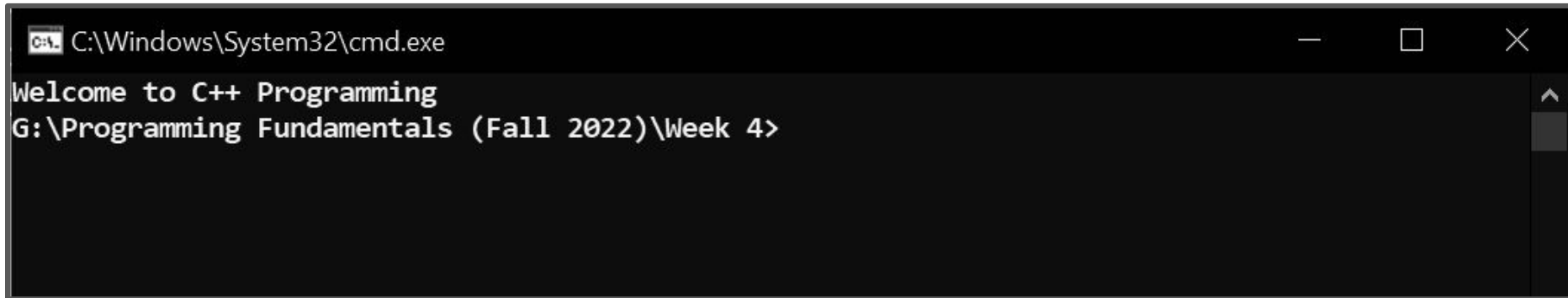
Activity 01: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);
main()
{
    system("cls");
    gotoxy(0, 0);
    cout << "Welcome to C++ Programming";
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```


Activity 01: Solution

First of all everything will be cleared from the console and then “Welcome to C++ programming” will be written at (0,0) coordinates of the console.

A screenshot of a Windows Command Prompt window. The title bar at the top reads "C:\Windows\System32\cmd.exe" and includes standard window control buttons (minimize, maximize, close). The command prompt shows the output "Welcome to C++ Programming" on the first line. The second line shows the current directory path "G:\Programming Fundamentals (Fall 2022)\Week 4>". The text is white on a black background. A vertical scrollbar is visible on the right side of the window.

```
C:\Windows\System32\cmd.exe
Welcome to C++ Programming
G:\Programming Fundamentals (Fall 2022)\Week 4>
```

Activity 02

Let's write "Welcome to C++ Programming" at the (5,0) coordinates of console.



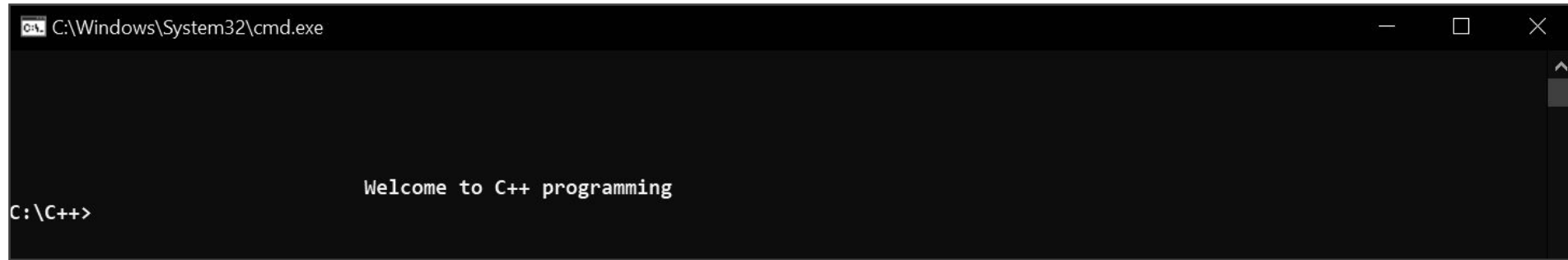
```
C:\Windows\System32\cmd.exe
Welcome to C++ programming
C:\C++>
```

Activity 02: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);
main()
{   system("cls");
    gotoxy(5, 0);
    cout << "Welcome to C++ Programming";
}
void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 03

Let's write "Welcome to C++ Programming" at the 30th x and 5th y (30,5) coordinates of console



```
C:\Windows\System32\cmd.exe

Welcome to C++ programming

C:\C++>
```

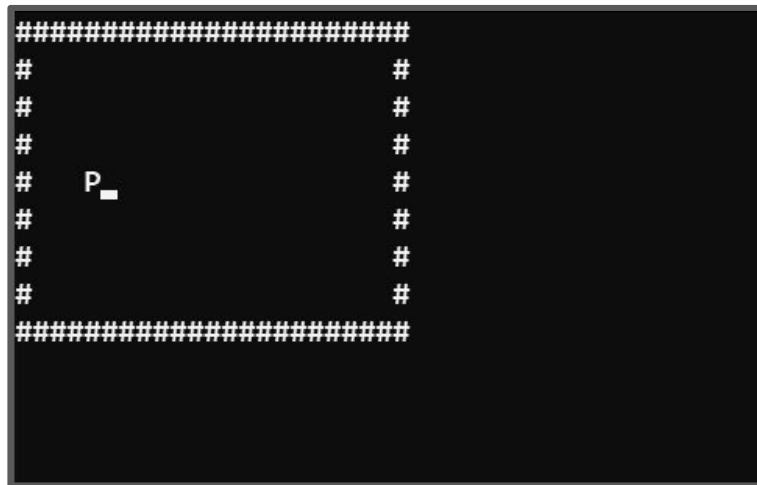
The image shows a Windows Command Prompt window with a black background and white text. The title bar at the top reads 'C:\Windows\System32\cmd.exe'. The main area of the window displays the text 'Welcome to C++ programming' at the 30th column and 5th row. The prompt 'C:\C++>' is visible at the bottom left of the window.

Activity 03: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);
main()
{   system("cls");
    gotoxy(30, 5);
    cout << "Welcome to C++ Programming";
}
void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

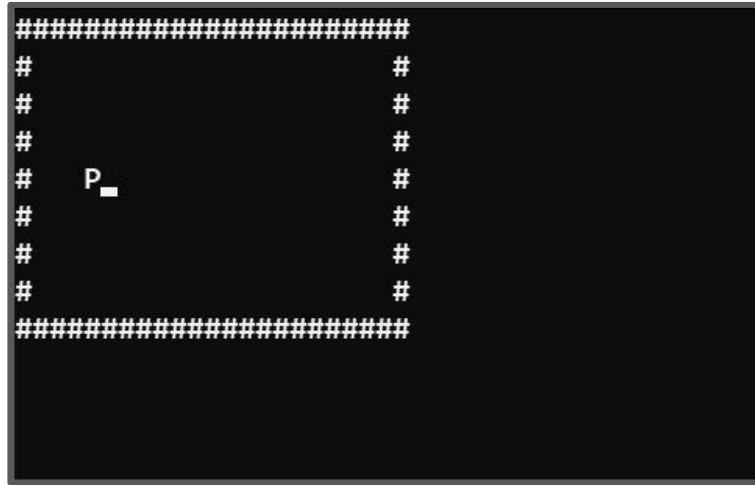
Goal: Move Pacman on the Console

Now, let's come back to the goal of this lecture.



|| Goal: Move Pacman on the Console

Lets simply make a function to print the maze first.



Goal: Move Pacman on the Console

```
#include <iostream>
#include <windows.h>
using namespace std;

void gotoxy(int x, int y);
void printMaze();

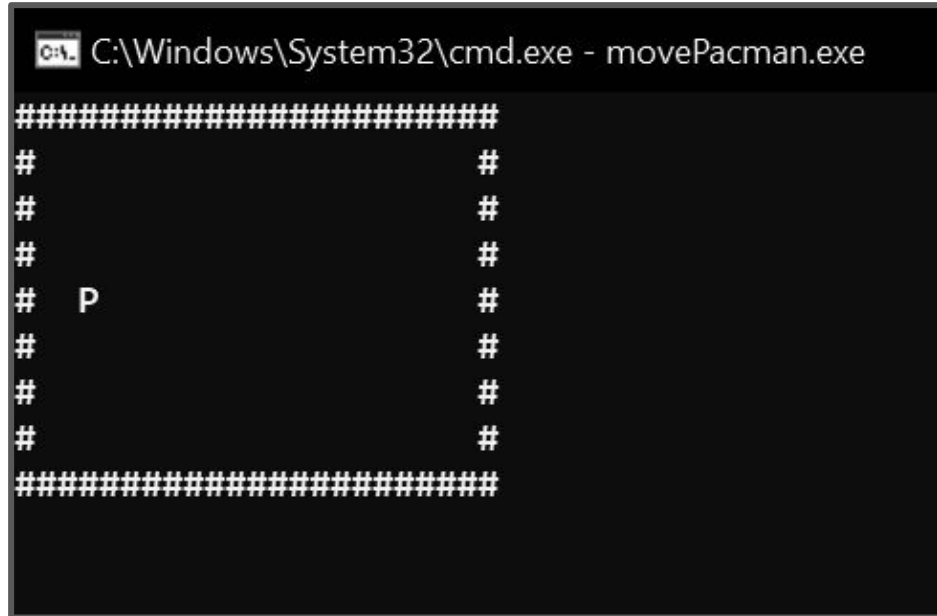
main()
{
    system("cls");
    printMaze();
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```


Goal: Move Pacman on the Console

Lets now place the pacman at (3, 4) coordinate.



```
C:\Windows\System32\cmd.exe - movePacman.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
#                                     #
#                                     #
#                                     #
#####
```

Goal: Move Pacman on the Console

```
#include <iostream>
#include <windows.h>
using namespace std;

void gotoxy(int x, int y);
void printMaze();

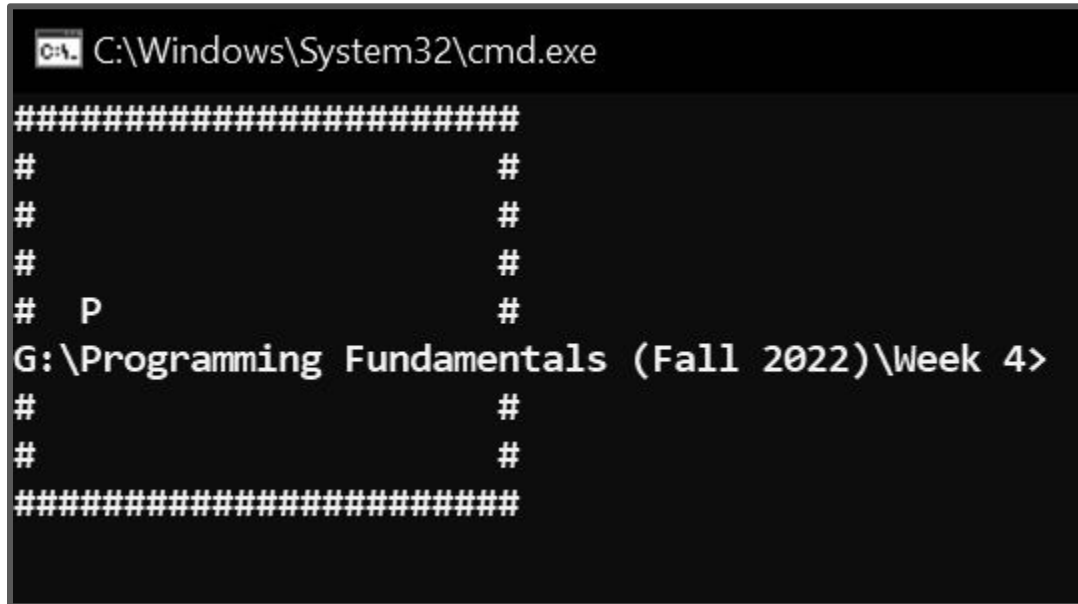
main()
{
    system("cls");
    printMaze();
    gotoxy(3,4);
    cout << "P";
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Goal: Move Pacman on the Console

After Compiling and executing the program, the output becomes.

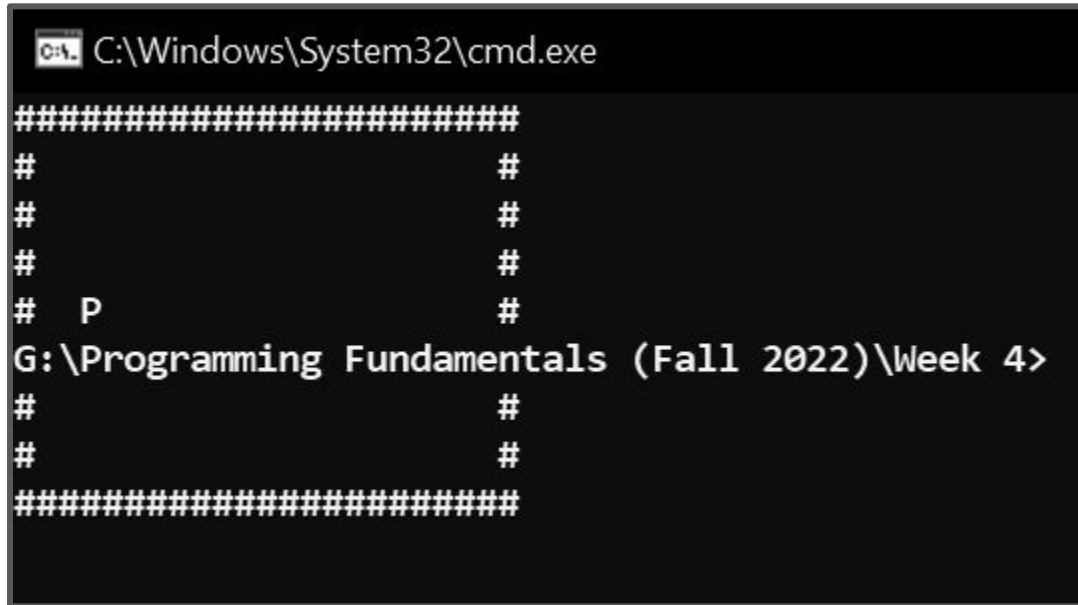


```
C:\Windows\System32\cmd.exe

#####
#                                     #
#                                     #
#                                     #
#   P                               #
G:\Programming Fundamentals (Fall 2022)\Week 4>
#                                     #
#                                     #
#####
```

Goal: Move Pacman on the Console

Since the program terminates, therefore, the text is overlapped.

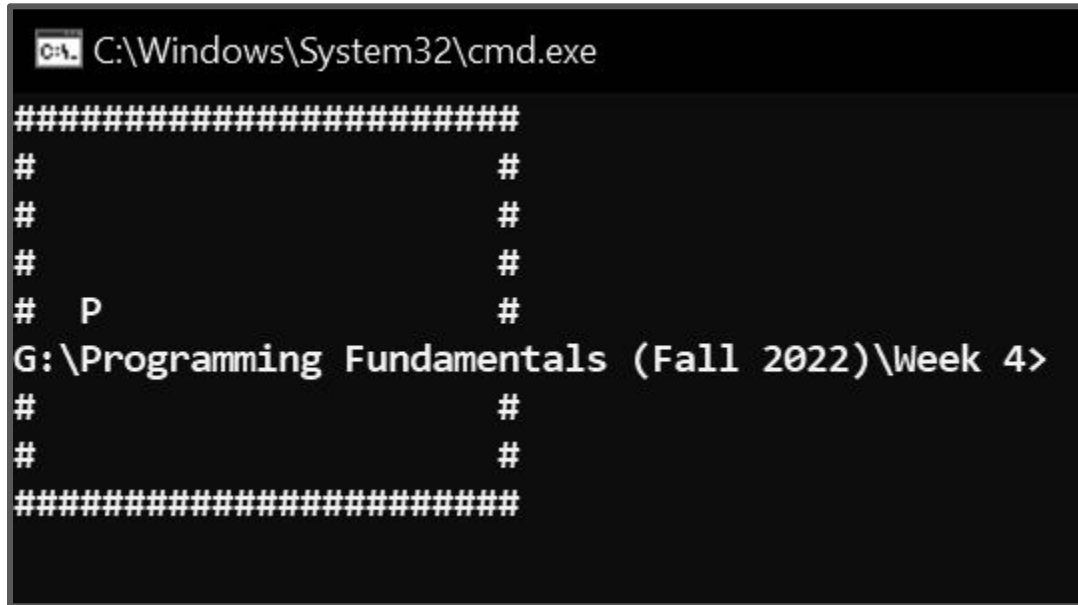


```
C:\Windows\System32\cmd.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
G:\Programming Fundamentals (Fall 2022)\Week 4>
#                                     #
#                                     #
#####
```

Goal: Move Pacman on the Console

Now, we want that the program keeps running.

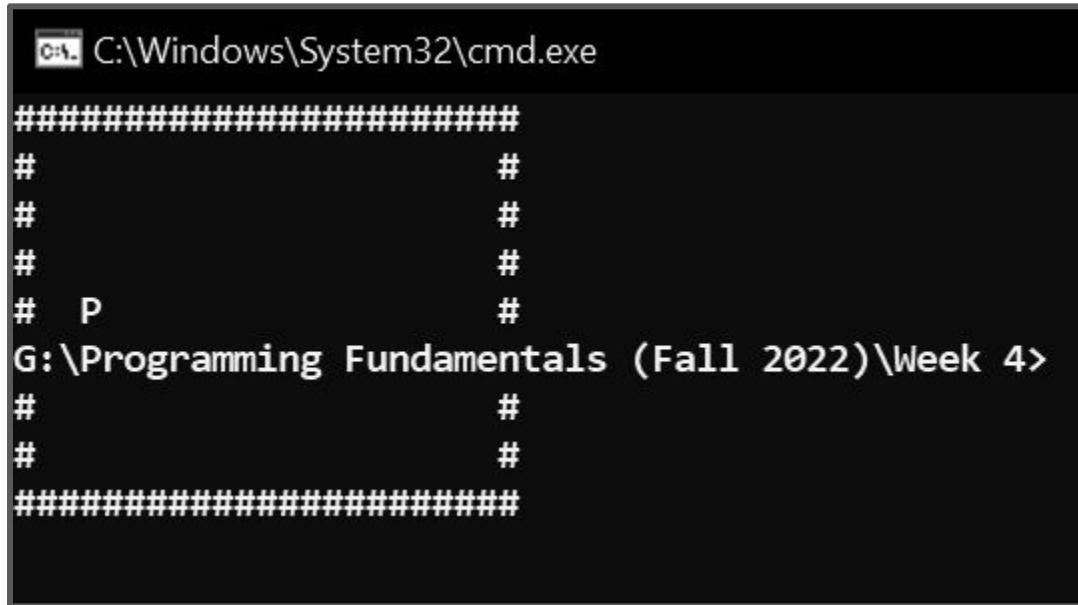


```
C:\Windows\System32\cmd.exe

#####
#                                     #
#                                     #
#                                     #
#   P                               #
G:\Programming Fundamentals (Fall 2022)\Week 4>
#                                     #
#                                     #
#####
```

Goal: Move Pacman on the Console

How, can we do that?

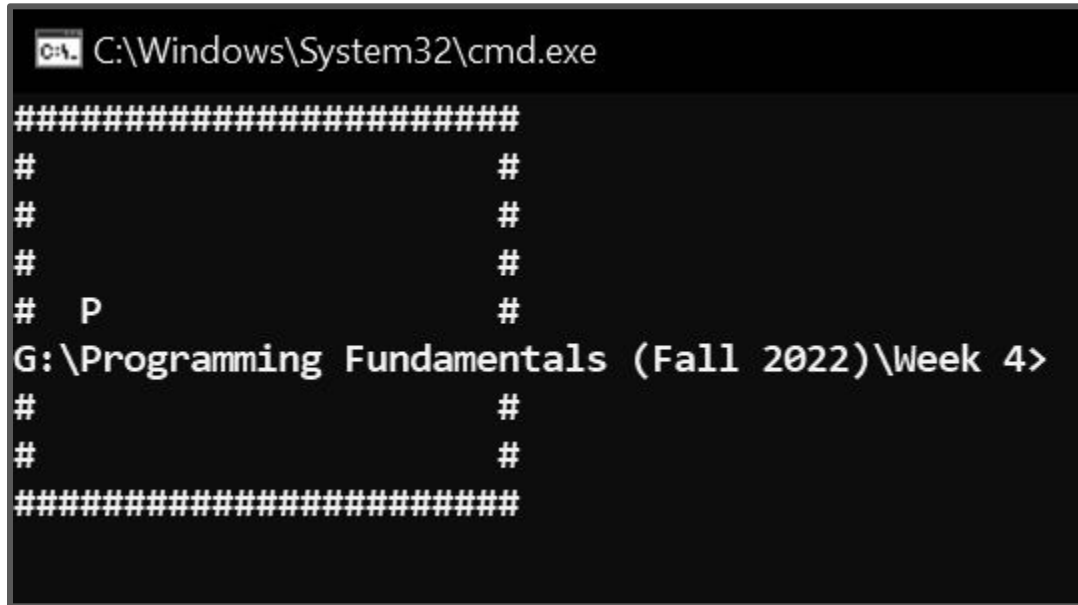


```
C:\Windows\System32\cmd.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
G:\Programming Fundamentals (Fall 2022)\Week 4>
#                                     #
#                                     #
#####
```

Goal: Move Pacman on the Console

We have to place the code in the **while** loop.

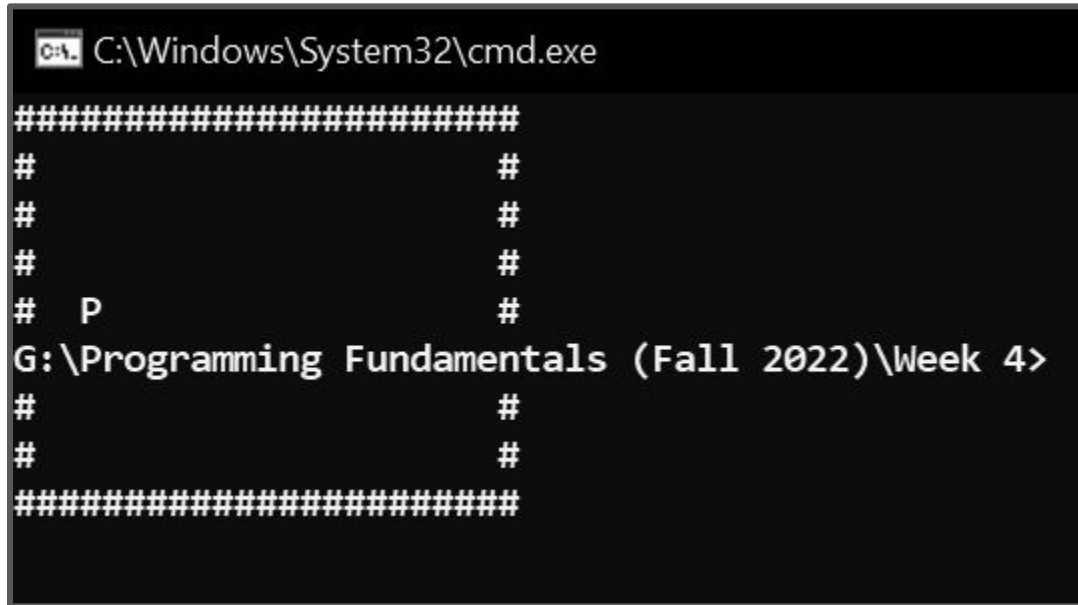


```
C:\Windows\System32\cmd.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
G:\Programming Fundamentals (Fall 2022)\Week 4>
#                                     #
#                                     #
#####
```

Goal: Move Pacman on the Console

Let's keep printing the pacman on the same coordinates.



```
C:\Windows\System32\cmd.exe

#####
#                                     #
#                                     #
#                                     #
#  P                                 #
G:\Programming Fundamentals (Fall 2022)\Week 4>
#                                     #
#                                     #
#####
```


Goal: Move Pacman on the Console

```
#include <iostream>
#include <windows.h>
using namespace std;

void gotoxy(int x, int y);
void printMaze();

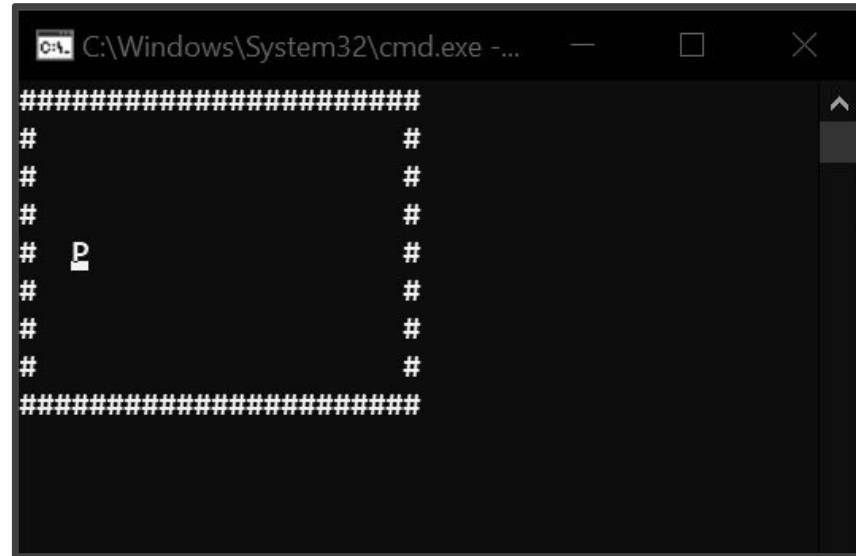
main()
{
    system("cls");
    printMaze();
    while (true)
    {
        gotoxy(3, 4);
        cout << "P";
    }
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Goal: Move Pacman on the Console

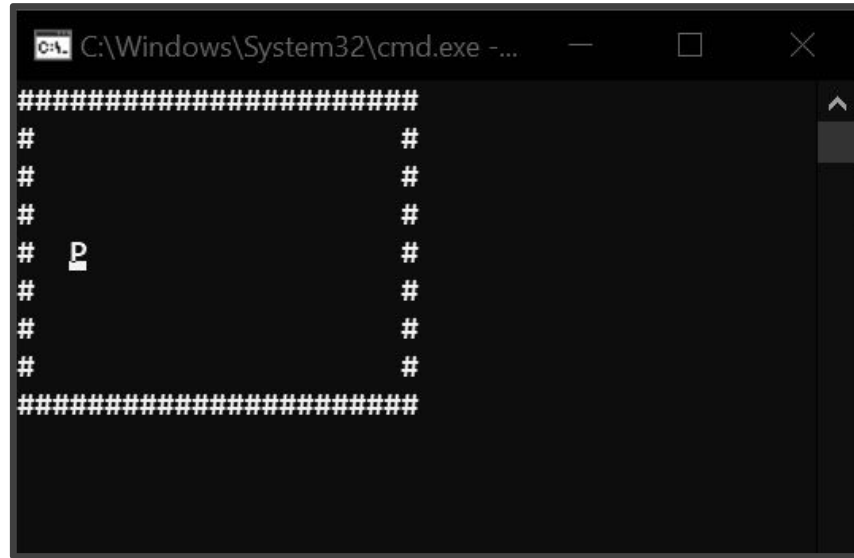
Now, the program is running continuously, and the pacman is keep on printing on the (3,4) coordinates.



```
C:\Windows\System32\cmd.exe -...  
#####  
#                                     #  
#                                     #  
#                                     #  
#  P                                 #  
#                                     #  
#                                     #  
#                                     #  
#####
```

Goal: Move Pacman on the Console

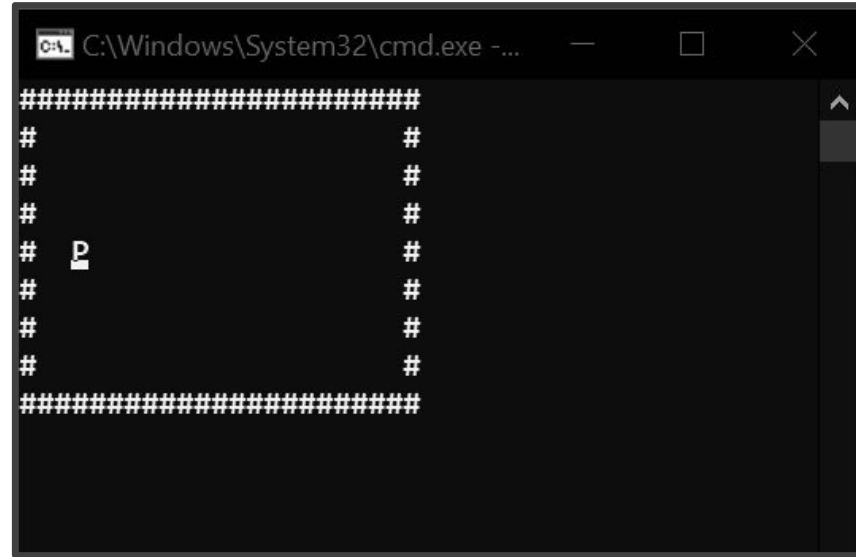
We have to move the pacman to the right side i.e., we have to **update the x coordinate** in the gotoxy function



```
C:\Windows\System32\cmd.exe -...  
#####  
#                                     #  
#                                     #  
#                                     #  
# P                                  #  
#                                     #  
#                                     #  
#                                     #  
#####
```

Goal: Move Pacman on the Console

Let's make the variables of x and y and then pass these variables in the **gotoxy** function.



```
C:\Windows\System32\cmd.exe -...  
#####  
#                                     #  
#                                     #  
#                                     #  
#  P                                 #  
#                                     #  
#                                     #  
#                                     #  
#####
```

Goal: Move Pacman on the Console

```
#include <iostream>
#include <windows.h>
using namespace std;

void gotoxy(int x, int y);
void printMaze();

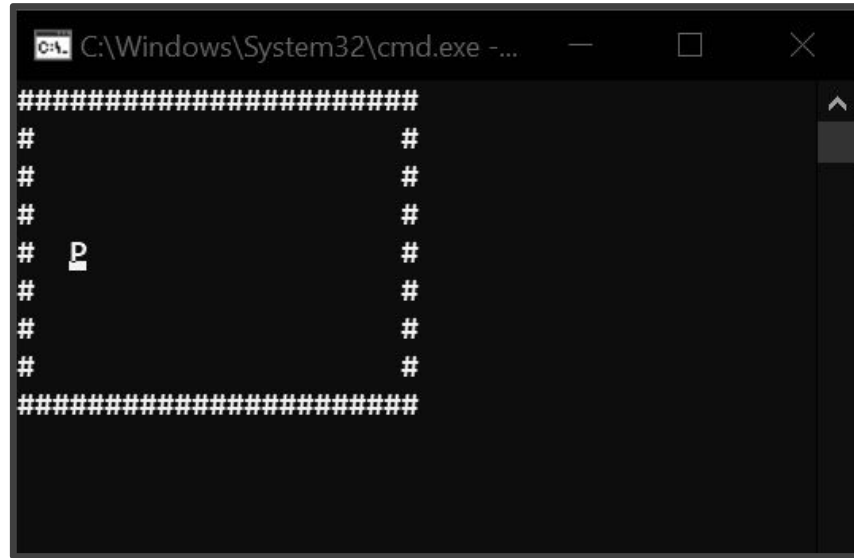
main()
{
    system("cls");
    printMaze();
    int x = 3;
    int y = 4;
    while (true)
    {
        gotoxy(x, y);
        cout << "P";
    }
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Goal: Move Pacman on the Console

Now, in order to move the pacman towards right, we have to update the **x variable by 1**.



```
C:\Windows\System32\cmd.exe -...  
#####  
#                                     #  
#                                     #  
#                                     #  
#  P                                 #  
#                                     #  
#                                     #  
#                                     #  
#####
```

Goal: Move Pacman on the Console

```
#include <iostream>
#include <windows.h>
using namespace std;

void gotoxy(int x, int y);
void printMaze();

main()
{
    system("cls");
    printMaze();
    int x = 3;
    int y = 4;
    while (true)
    {
        gotoxy(x, y);
        cout << "P";
        x = x + 1;
    }
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Goal: Move Pacman on the Console

After **Compiling and Executing** the program, the output becomes.

[illegible]

Goal: Move Pacman on the Console

It printed P on the (3, 4) coordinate, then on (4,4), then on (5,4) and keep on printing on the updated coordinates.

[illegible]

Goal: Move Pacman on the Console

Now, we want to print the pacman only till it reaches to (20, 4) coordinate then it should start back from the (3, 4).

[illegible]

Goal: Move Pacman on the Console

How can we do that?

[illegible]

Goal: Move Pacman on the Console

We can apply the condition that if the Pacman reaches at `x == 20` then we should start the x with 3 again.

[illegible]

Goal: Move Pacman on the Console

```
#include <iostream>
#include <windows.h>
using namespace std;
void gotoxy(int x, int y);
void printMaze();
main()
{
    system("CLS");
    printMaze();
    int x = 3;
    int y = 4;
    while (true)
    {
        gotoxy(x, y);
        cout << "P";
        x = x + 1;
        if(x == 20)
        {
            x = 3;
        }
    }
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Goal: Move Pacman on the Console

Now, the output becomes.

```
#####  
#                                     #  
#                                     #  
#                                     #  
#  P P P P P P P P P P P P P P P P #  
#                                     #  
#                                     #  
#                                     #  
#####
```

Goal: Move Pacman on the Console

Now the only issue is that once the P is printed at some coordinate then it is placed there permanently.



Goal: Move Pacman on the Console

But, we want to place the empty space on the previous location of the Pacman once its x coordinate is updated.



Goal: Move Pacman on the Console

```
#include <iostream>
#include <windows.h>
using namespace std;
void gotoxy(int x, int y);
void printMaze();
main() {
    system("CLS");
    printMaze();
    int x = 3;
    int y = 4;
    while (true)
    {
        gotoxy(x, y);
        cout << " ";
        x = x + 1;
        if(x == 20)
        {
            x = 3;
        }
        gotoxy(x, y);
        cout << "P";
    }
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Goal: Move Pacman on the Console

Now, after execution the output becomes.

```
#####  
#                                     #  
#                                     #  
#                                     #  
#      P_                            #  
#                                     #  
#                                     #  
#                                     #  
#                                     #  
#####
```

Goal: Move Pacman on the Console

Now, the working is perfectly fine, but the pacman is moving very fast.



Goal: Move Pacman on the Console

We have to let the code to stop for some seconds then the pacman should move on the next coordinate.



Goal: Move Pacman on the Console

In C++, if we want the code to not execute for some time, we have the following function `Sleep(200)`, its input parameter is the time in milliseconds for which the code should not execute.



Goal: Move Pacman on the Console

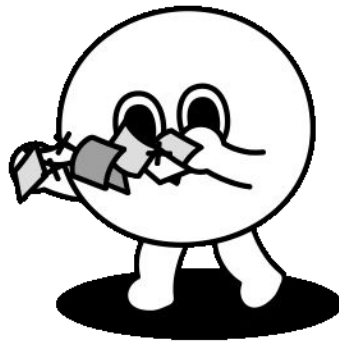
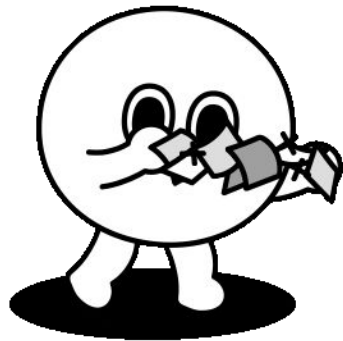
```
#include <iostream>
#include <windows.h>
using namespace std;
void gotoxy(int x, int y);
void printMaze();
main() {
    system("CLS");
    printMaze();
    int x = 3;
    int y = 4;
    while (true)
    {
        gotoxy(x, y);
        cout << " ";
        x = x + 1;
        if(x == 20)
        {
            x = 3;
        }
        gotoxy(x, y);
        cout << "P";
        Sleep(200);
    }
}
```

```
void printMaze()
{
    cout << "#####" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#                #" << endl;
    cout << "#####" << endl;
}

void gotoxy(int x, int y)
{
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Goal: Move Pacman on the Console

Finally, the goal is achieved.



Learning Objective

Write a **C++ program** to display output on the console at a specific location using **gotoxy()** function.



Self Assessment

You have to convert your **Vision**, to concrete requirements. Following is the sample to write your requirements for your Business Application.

User Story ID	As a	I want to perform	So that I can
1	Admin	Calculate the Aggregate.	Calculate the aggregate of the students w.r.t. their fsc and ecat marks.

