



**Foundation Certificate for Higher Education**

**Module code:** DOC-333-Introduction to Programming in Python – P1

**Module Leader:** Mr.Sudarshan Welihinda

**Assignment Number:** 1

**Assignment Type:** Individual Coursework

**Title:** Developing the Hidden Peg Game in Python

**Student Name:** H.H.Welgama

**Student ID:** 20221109

## Contents

Table of figures .....	2
1 Introduction.....	2
1.1The Primary purpose of the game concept.....	2
1.2    The Python Program .....	2
2 How does the program operate?.....	3
3 Python code of the program .....	3
4 Screenshots of the program.....	7
5 Conclusion .....	8

## Table of figures

Figure 1:Flowchart .....	3
Figure 2: Program output that shows the user finding the answer.....	8
Figure 3: program output that shows the user not finding the answer .....	8

## 1 Introduction

This report is a brief overview of the Python program created to implement the code breaking game concept called “The Hidden Peg Game”.

### 1.1The Primary purpose of the game concept

In this game, two players take turns being the code writer and code breaker. The code maker uses 6 color to build a 4-digit code, the code breaker uses 6 colors to break that code, and the code breaker uses 6 colors to generate a 4. - Numeric code. The encoder will give you a shot of a black peg, indicating that the colored peg is the correct color and in the correct position, and a white peg, indicating that the colored peg is the correct color but in the displayed position. The code breaker has 8 chances to figure out this code.

### 1.2 The Python Program

The main idea of the power game above is taken from the Python software mentioned in this report to create an integer-based code-breaking game. The color palette of this software is represented by integers from 0 to 6, and white and color black are represented by the numbers 0

and 1 respectively. The program's code generator generates a four-digit code, takes input from the user acting as a code breaker in this case, and then provides a response tip. The user has 8 attempts to crack the system generated code, just like in the game. If the chances have run out, the program will end, giving the chance for the user to restart the game.

## 2 How does the program operate?

Conditional statements are primarily used in this software to see if the user inputs match the code that needs to be located and to deliver the appropriate hints so that the user may take the appropriate actions. Using a flowchart diagram to illustrate the operation of the previously mentioned program will help you understand this better.

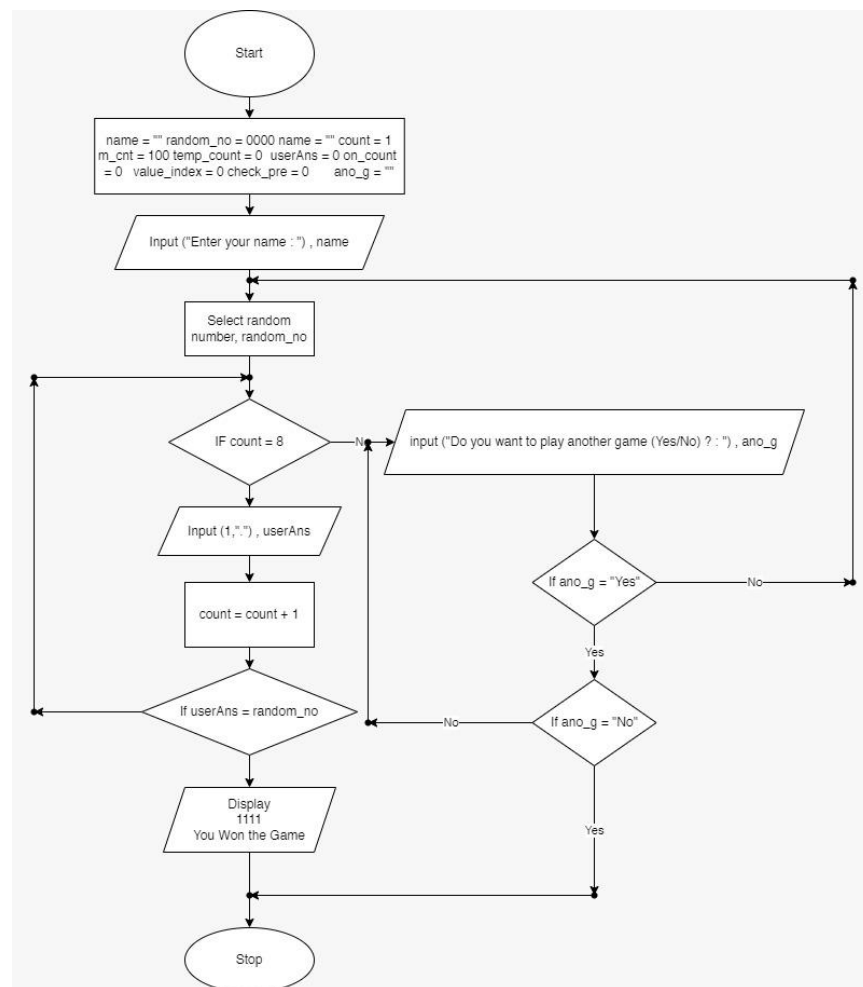


Figure 1:Flowchart

## 3 Python code of the program

```
import random as gen
```

```
random_no = str(gen.randint(1, 6)) + str(gen.randint(1, 6)) + str(gen.randint(1, 6)) + str(gen.randint(1, 6))
```

```
print("\t\t\t HI", name, "Welcome to GameInt")
```

```
print()
```

```
print("\t\t\t\t\t\t\tColor Mapping: ")
```

[illegible][illegible]

```
print()
```

```
ans = []
```

```
def game():
```

```
count = 1 #count untill 8th attempts
```

```
m_cnt = 100
```

```
print("Attempt\t\t\t\t\t Guess\t\t\t\t\t Result ")
```

while 1:

```
print("_____")
```

while True:

```
temp_count = 0
```

```

userAns = input(f"{count}: \t\t\t\t\t ")
on_count = 0
for index in userAns:
    if index == '0':
        on_count += 1
if on_count == 4:
    print()
    print(".....Terminate Game Int....")
    exit()
    break
for index in userAns:
    temp = int(index)
    if 0 < temp < 7:
        temp_count += 1
if temp_count == 4 and len(userAns) == 4:
    break
else:
    print()
    print("!!!!Wrong Input Please Enter Again.....")
value_index = 0
check_pre = 0
for index in userAns:
    check = True
    if random_no[value_index] == index:
        ans.insert(value_index, '1') # check 1 val
        check_pre += 1
        check = False
    else:

```

```

if check_pre > 1:
    for k_val in range(check_pre, 4):
        if index == random_no[k_val]:
            ans.insert(value_index, '0')
            check = False
    else:
        for f in random_no:
            if index == f:
                ans.insert(value_index, '0')
                check = False
                break
    if check:
        ans.insert(value_index, '.')
    value_index += 1
print("\t\t\t\t\t\t\t\t\t\t", end=' ')
win = 0
for v in range(0, 2):
    print(ans[v], end=" ")
print()
print("\t\t\t\t\t\t\t\t\t\t", end=' ')
for v in range(2, 4):
    print(ans[v], end=" ")
for c_val in ans:
    if c_val == '1':
        win += 1
if win == 4:
    print()
    print("Congratulations !!!!! You have won the game...")

```



[illegible]

## 5 Conclusion

DOC-333-Introduction to Programming in Python