Problem Solving using c and c++

Lab Assignment 3

Patil Amit Gurusidhappa
19104004
B11

1. Why is it unsafe using gets() in C/C++? Write a C/C++ program to demonstrate the same.
Further, rewrite the program using gets() alternative.

Ans.

gets() reads from stdin, it keeps on reading until the new line character is found or end of file. So in this case the user can keep on giving input which eventually may result in buffer overflow. So gets() suffers from the BufferOverflow Problem.

```
#include <stdio.h>

void read()
{
char str[20];
gets(str);
printf("%s", str);
return;
}


int main() {
        read();
        return 0;
}
```

Alternative code

We can use fgets() in place of gets() as it takes input until new line character or the buffer size

```
#include <stdio.h>

void read()
{
char str[20];
fgets(str,20,stdin);
printf("%s", str);
return;
}


int main() {
        read();
        return 0;
}
```

2. Analyze the following code fragment for a string whose length is more than 10 bytes and find
the error in code. What could be the possible reasons for the error and how do we rectify it?

3. A buffer, in terms of a program in execution, can be thought of as a region of computer’s main
memory that has certain boundaries in context with the program variable that references this memory. A buffer is said to be overflown when the data (meant to be written into memory buffer) gets written past the left or the right boundary of the buffer. This way the data gets written to a portion of memory which does not belong to the program variable that references the buffer. Due to this, a program could crash or give unexpected results. Buffer overflow also leads to buffer overflow attacks. Write a C program to demonstrate buffer overflow. Also, discuss the prevention strategies to avoid buffer overflow attacks.

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char buff[15];
    int pass = 0;

    printf("\n Enter the password : \n");
    gets(buff);

    if(strcmp(buff, "thegeekstuff"))
    {
        printf ("\n Wrong Password \n");
    }
    else
    {
        printf ("\n Correct Password \n");
        pass = 1;
    }

    if(pass)
    {
       /* Now Give root or admin rights to user*/
        printf ("\n Root privileges given to the user \n");
    }

    return 0;
}
```

4. What will be the output of the following code?

OUTPUT

```
i = -2147483648, j = 2147483647, k = -2147483648, u = 0, v = 4294967295
```