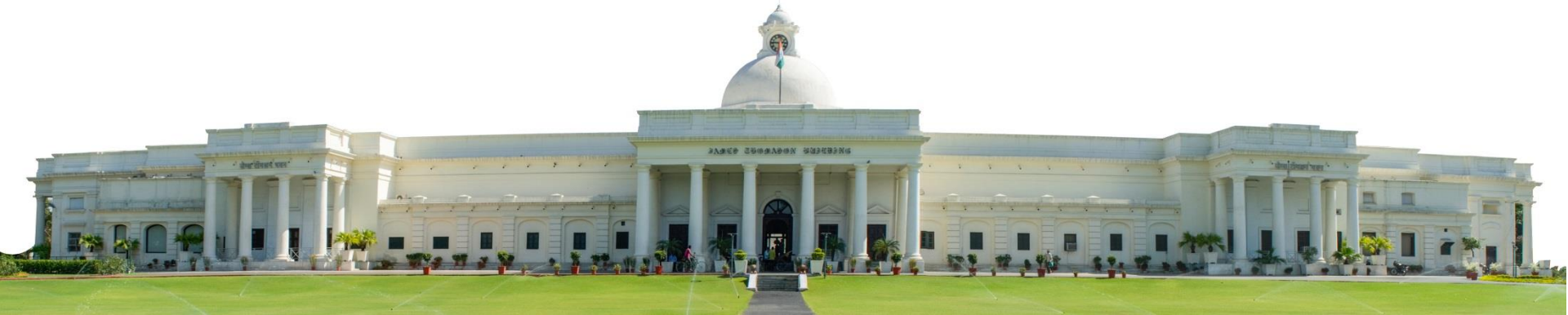
Programming with C and C++

CSC-101 (Lecture 22)

Dr. R. Balasubramanian
Professor
Department of Computer Science and Engineering
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology Roorkee
Roorkee 247 667

bala@cs.iitr.ac.in
https://faculty.iitr.ac.in/cs/bala/



Palindrome using function

</> source code

https://ideone.com/Y5Y3Zq

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

bool isPalindrome(const char *str) {
    int len = strlen(str);
    int i, j;

    for (i = 0, j = len - 1; i < j; i++, j--) {
        if (str[i] != str[j])
            return false;
    }

    return true;
}

```

```c
int main() {
    char inputString[100];
    printf("Enter a string or number to check for palindrome: ");
    scanf("%s", inputString);

    if (isPalindrome(inputString))
        printf("%s is a palindrome.\n", inputString);
    else
        printf("%s is not a palindrome.\n", inputString);

    return 0;
}

```

stdout

```
Enter a string or number to check for palindrome: malayalam is a palindrome.
```

Character Arrays and Pointers

```c
#include<stdio.h>

void printString(char *str)
{
    if(*str == '\0')
    {
        return;
    }
    printf("%c", *str);
    printString(++str);
}

int main()
{
    char str[15] = "Recursed";
    printString(str);
    return 0;
}

```

https://ideone.com/N5f148

stdout

```
Recursed
```

```c
#include <stdio.h>

int main(void) {
    char c[ ] = "CSE2023";
    char *p = c;
    printf("%s", p + p[2] - p[0] + 1);
    return 0;
}
```

stdout

```
2023
```

https://ideone.com/YHhF1n

Static variable

- An ordinary variable is limited to the scope in which it is defined, while the scope of the static variable is throughout the program.

Static variable

```c
#include <stdio.h>
int main()
{
    printf("%d",func());
    printf("\n%d",func());
    return 0;
}

int func()
{
    int count=0; // variable initialization
    count++; // incrementing counter variable
    return count;
}
```

stdout

```
1
1
```

https://ideone.com/kibY8C

Static variable

```c
#include <stdio.h>
int main()
{
    printf("%d",func());
    printf("\n%d",func());
    return 0;
}

int func()
{
    static int count=0; // static
    count++; // incrementing counter variable
    return count;
}
```

stdout

```
1
2
```

https://ideone.com/sgUF2z

Static variable

```c
#include <stdio.h>

int f(int n)
{
    int r = 0;
    if(n <= 0) return 1;
    if(n>3)
    {
        r = n;
        return f(n-2)+2;
    }
  return f(n-1)+r;
}

```

Static variable

```c
int main(void) {
    // your code goes here
    int k=f(5);
    printf("%d",k);
    return 0;
}
```

stdout

3

https://ideone.com/InocVe

Find the output?

```c
#include <stdio.h>
int main()
{
    int a[] = {2, 4, 6, 8, 10} ;
    int i, sum = 0, *b = a + 4 ;
    for (i = 0; i < 5; i++)
        sum = sum + (*b - i) - *(b - i) ;
    printf ("%d\n", sum) ;
    return 0 ;
}
```

Success #stdin #stdout 0s 5432KB

```
10
```

https://ideone.com/zmpESs

Static Array

```c
#include <stdio.h>

int main(void) {
	// your code goes here
	int a[10000000000];
	printf("%x",a);
	return 0;
}
```

https://ideone.com/A8PdXT

```
~$ ./a.out
Segmentation fault (core dumped)
~$
```

Runtime error #stdin #stdout 0s 5436KB

Dynamic Array

dynarr.c

```c
#include <stdio.h>
#include <stdlib.h>
int main(void) {
	// your code goes here
	//int a[10000000];
  int size = 10000000000;
    int *a = (int*) malloc(size * sizeof(int));
    	printf("%x\n",a);
	return 0;
}
```

```
~$ ./a.out
b43fc010
~$
```

stdout

```
ac398010
```

https://ideone.com/vqVO4I

Dynamic Array

- Dynamic arrays are a powerful data structure in programming that allows for creating and manipulating arrays of varying sizes during runtime.
- In C, the most commonly used memory allocation functions are **`malloc()`**, **`calloc()`**, and **`realloc()`**.

Dynamic Array

```c
#include <stdio.h>
#include <stdlib.h>
int main() {
    int size = 5;
    int *arr = (int*) malloc(size * sizeof(int));
    int i;

        for(i = 0; i< size; i++) {
        arr[i] = i;
        }
    // Add a new element to the array
        size++;
        arr = (int*) realloc(arr, size * sizeof(int));
        arr[size-1] = i;

```

https://ideone.com/0bf596

Dynamic Array

```
    for(i = 0; i< size; i++) {
        printf("%d ", arr[i]);
    }

    free(arr);

    return 0;
}
```

stdout

```
0 1 2 3 4 5
```

Dynamic Array

```c
#include <stdio.h>
#include <stdlib.h>
int main() {
    int size = 5;
    int *arr = (int*) malloc(size * sizeof(int));
    int i;

        for(i = 0; i< size; i++) {
        arr[i] = i;
        }
    // Add a new element to the array
        size++;
        arr = (int*) realloc(arr, size * sizeof(int));
        arr[size-1] = i;

```

https://ideone.com/4U59Q7

Dynamic Array

```
    for(i = 0; i< size; i++) {
        printf("%d ", arr[i]);
    }

    free(arr);

    printf("\n");

    for(i = 0; i< size; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

stdout

```
0 1 2 3 4 5
0 0 -1289768944 21846 4 5
```

Thank You!