Lec - 6

05/Feb/2023

(Functions and Problem Statement)

- **Functions** → A program linked with a well defined task.

Eg→ Print your name n times.

```
int n;
cin>>n;
for (int i=0; i<n; i++){
        cout << "Deepti";
}
```

suppose we want to print more time.
→ we again wrote this (copy/paste).

In this method code becomes bulky and lengthy.

→ No readability as it is so lengthy for such a small task.

→ If there is some mistake in this code then we have correct it again and again (where-ever we wrote it.) So code it Buggy.

A better way to create function and call that function whenever needed.

```
void print (){
    int n;
    cin>>n;
    for( int i=0 ; i<n; i++){
            cout << "Deepti";
    }
}
```

```
int main() {
    print();
}
```

Syntax -

```
return_type   functionName (   )
{
    - - -
    - - -     } func body.
    - - -
}
```

return_type ← eg→ int, void, float etc.

functionName ← we should give logical / meaningful names.

( ) ← Input parameters (seperated by commas)

```
eg→  void   printName ( ) {
         - - } → body.
     }
```

void ← return type

printName ← Name of the func.

( ) ← here we do not have any input parameter.

- int main()
  {
  - - .
  -
  return 0;
  }

(return type → int)

return 0; → This means successful execution of the function.

⊙ **Explore** →

→ Can we write more than 1 main() functions?
→ Can we pass i/p parameter in " . ?
→ What happens when we return 1 (+ve integer)?
→ What happens when we return -ve integer?

→ **Function** - Well-defined task

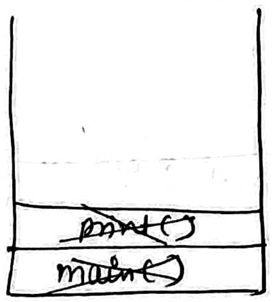
Advantages -
→ Reusability
→ Readable
→ ~~Re~~

⊙ **Function call stack** -

stack - As of now stack is a data structure (data is stored in a specific way)
→ follows LIFO (Last In First Out).

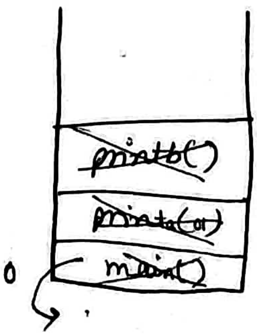
- → Function call stack tells the called function.
→ other fuctions they called.
→ return type, variables.

function call stack →




```
int main(){          → print(){
   print()              -
}  return0;             -
                      }
```

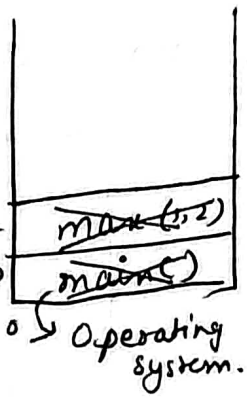
```
int main(){        [5 a]      void printa(int a) [3 b]     printb(int b){
   int a=5;                   {                              cout << b;
   printa(a);                    int b = 3;                }
   return 0;                     cout << a;
}                                printb(b);
                              }
```



```
main(){                          int max(int a, int b){
    int a=1                          if(a>b)
    int b=2                              return a;
    cout << max(a,b);                else
    return 0;                            return b;
}                                    }
```

a [1] b [2]

2



⊙ Whom main is returning 0 ?

⇒ OS (Operating system).

Search -

- What does return 0 means in C++ ?
- " " " 1 " " " ?
- " " " -1 " " " ?

⊙ **Pass By Value →**

①

```
int main(){
    int a=5;
    printNum(a);
    cout << a;
    return 0;
}
```

let add = 100, a = 5

```
void printNum(int a){
    cout << a;
    a++;
    cout << a;
}
```

Name of this variable can be different also.

o/p = 5

o/p → 6

copy of a is created.

104 ← address.

a = 5 6

eg →

```
main(){
    int num = 12;
    num++;
    ++num;
    print(num);
    --num;
    cout << num;
}
```

address = 312, num = 12 13

```
void print(int num){
    ++num;
    num++;
    cout << num;
    --num;
}
```

address = 416, num = 14 15 16

o/p = 16

o/p → 13

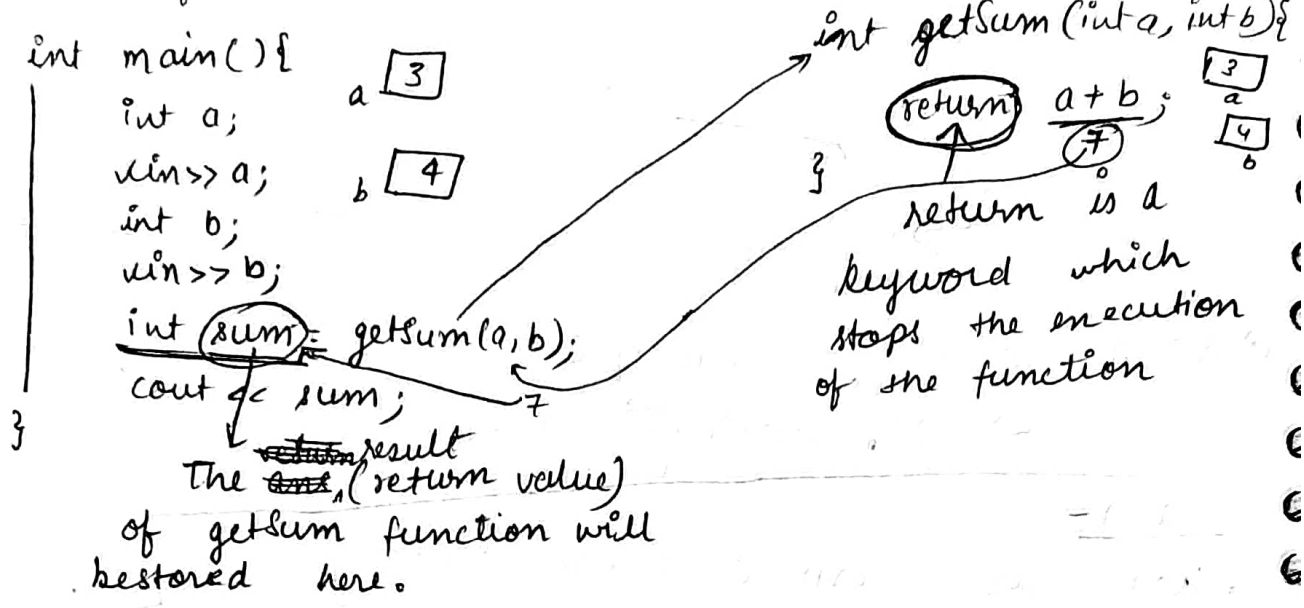
This is pass by value. (means copy of variable is passed not the actual variable).

- How to find address of a variable ?
  → By addressof operator (&).

```
int a;
cout << "address of a:" << (&a) << endl;
```

o/p → 0x2535abcd ← some hexadecimal value.

① Write a function to add 2 numbers.



⊙ We have to make sure that function declaration (atleast) should be above the place we are invoking that function.

- function declaration / Signature → It include return type, name and input parameters of the function.

eg → int add (int x, int y)

- function Definition →

eg → int add (int x, int y){
return x+y;
}

After declaration the function we can define it anywhere. (Before or after main() function).

② Find max of 3 numbers -

```
int main()
{
    int a, b, c;
    cin>>a>>b>>c;
int maximumNumber = findMax(a,b,c);
cout << "Maximum Number is " <<
    maximumNumber << endl;
}
```

```
int maximumNumber
(int num1, int num2,
 num3) {
 if (num1>num2 &&
    num1>num3)
    return num1;
 else if (num2>num1
  && num2>num3)
    return num2;
 else
    return num3;
}
```

⊙ Counting from 1 to n →

```
int main(){
    int n;
    cin>>n;
    printCounting(n);
}
```

```
void printCounting(int n){
    for(int i=1; i<=n; i++)
        cout<<i<<" ";
}
```

⊙ Function of Students & Grade Problem →

char
returntype → char

```
int main(){
    int marks;
    cin>>marks;
    char grade = getgrade(marks);
    cout<<grade;
}
```

marks → argument

```
char getgrade(int m){
    if(m>=90)
        return 'A';
    else if(m>=80)
        return 'B';
    else if(m>=70)
        return 'C';
    else if(m>=60)
        return 'D';
    else
        return 'E';
}
```

i/p → 94
o/p → A

i/p 86
o/p → B

i/p 44
o/p → E

⊙ Let's try using switch →

```
char getgrade(int m){
    switch(marks/10){
        case 10:
        case 9 : return 'A'; break;
        case 8: return 'B'; break;
        case 7: return 'C'; break;
        case 6: return 'D'; break;
        default : return 'E';
    }
}
```

⊙ Sum of even numbers upto N.

Let first write a program to find sum of all numbers from 1 to n.

```
int getSum (int n){
    int sum = 0;
    for (int i = 1; i <= n; i += 1)
        sum += i;
    }
    return sum;
}
```

```
main (){
    int n;
    cin >> n;
    cout << getSum (n);
}
```

i/p → 10
o/p → 55.

Now let's try to find only even number's sum.

```
int getEvenSum (int n){
    int evenSum = 0;
    for (int i = 0; i <= N; i += 1){
        evenSum += i;
    }
    return evenSum;
}
```

```
int main(){
    int n;
    cin >> n;
    cout << getEven
    Sum (n);
}
```

i/p → 10
o/p → 30

This can be done by if-else also.

```
for ( int i = 1; i <= n; i++){
    if (i % 2 == 0)
        evenSum += i;
}
```

But % (modulus operator) is very heavy operator. So avoid using it. To avoid this we can use Bitwise AND operator which will be discussed later.

Q.W →

- function to display area of circle
- Find number is even or odd
- Find factorial of a number
- Check Number is prime or not
- Print all prime number from 1 to n.

⑥ Problem Solving →