▶ functions.

↳ a block of code / sub-program.

↳ that is linked to a well-designed task.

↳ why → Resubile
↳ Readable.

issues (that's why we use this)

↳ lengthy / Bulky
↳ Buggy
↳ Readibility.
↳ Resue — X

```
                 → function name
→ int   main()          (entry point)
      {          ↳ input paramaters.
          ~~~~
return    ~~~~
type.     ~~~~
          return 0;
      }         ↳ says that program has
                  been successfully executed.
```

→ Simple function.

"Sundar ko sundari pasand hai"

Print 10 times

```
Void printLine (          )
{
    for (int i=0; i<10; i++)
    {
        Cout<<"Sundar ko sundari
                 pasand hai";
    }
}

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

function declaration.

```
Void PrintMessage();
int   addNumbers(int a, int b);
```

function definition.

```
int addno (int a, int b){
    return a+b;
}
```

→ Simple function.

"Sundar ko sundari pasand hai"

Print 10 times

```
void printLine (        )
{
    for (int i=0; i<10; i++)
    {
        Cout<<"sundar ko sundari
                pasand hai";
    }
}

int main(){

  Printline();

  return 0;

};
```

function declaration.

```
Void PrintMessage ();

int addNumbers (int a, int b);
```

function definition.

```
int addno (int a, int b){
    return a+b;
}
```

Functions call

FunctionName ( ); if you have arguments the put it here.

print-no (a,b);

Existance of function in c++.

↑ ~~att~~ function definition.

Line 110 - function call

you have to make sure that the definition of function is always above of calling function.

and if ~~eg~~ you want to run the function even from below then you have to declare the function at the top.

Snippet

```
~~void printLine(){~~
    ~~Cout<<"Hey! How are you"<<endl;~~
~~}~~

void printLine(); // declaration.
main(){
    printline(); // call.
  return 0;  }
void printLine(){ // definition.
    Cout<<"Hey There! How are you";
}.
```

functions have multiple return types which return value as of their data type but as we know void is a null data type so, void does not return anything.

Snippets

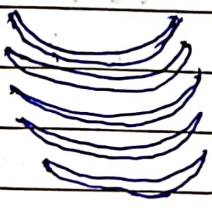
```
void printA(){
  cout<<"I am inside A"<<endl;
}

int main(){
   cout<<"Hi"<<endl;
    PrintA();

   return 0;
}
```

→ Function call stack.

↳ info about function/calls. Stack? (D.S)
↳ Local variables.
↳ multiple function callings. — stack of plates in marriages.
↳ Returing function of multiple data types.

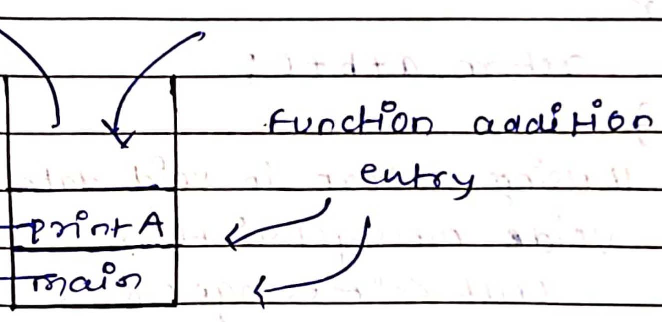


→ Last in First Out.

① Snippet.

```
Int main(){
  Cout<<"inside main";
  printA();
  Cout<<"Back in main";
  return 0;
}.

Void printA(){
  Cout<<"inside A";
  Cout<<"going back
         to main";
}.
```



removes from function call stack after implementing of functions.

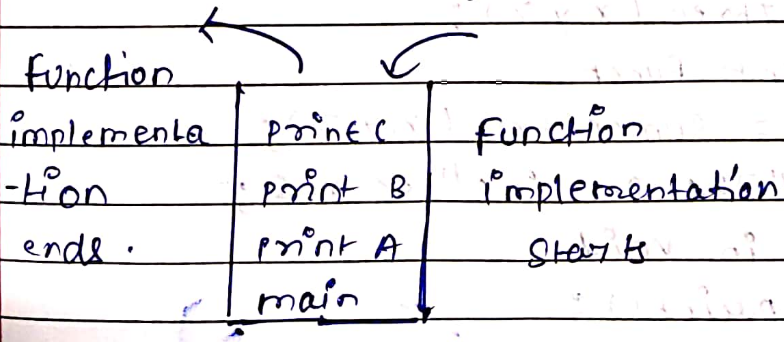
②

```
int main(){
  printA();
}.

printA(){
  Cout<<"insideA";
  Print B();
}

print B(){
  Cout<<"inside B";
  PrintC();
}

print C(){
  Cout<<"inside c";
}.
```


Function implementa-tion ends.
printC
print B
print A
main
Function implementation starts

Q. Sum of Three no's using function.

```
#include<iostream>
using namespace std;
// using void data type.
void sumofThree(int a, int b, int c)
{
  cout<<"The sum is :"<< a+b+c;
}

// using integer data type.
int sumofThree (int a, int b, int c)
{
   return a+b+c;
}
// using return in void data type func.
void messageprint(){
   cout <<"message one";
       return;
   cout <<"message two";
}

int main(){
int x,y,z;
cin>>x>>y>>z;
// for void func
  sumofThree (x,y,z);
// for int funct
int res = sumofThree (x,y,z);
cout<<"The sum is:"<< res;
// return in void
messageprint();
}
```