2.4 functions

Made By:-

Aditya Jain

function :- a block of code / sub-program
- that is linked to a well-designed task
- only run when it is called

why?
- Reusable
- Readable
- reducing code redundancy

Issues w/o function
- lengthy
- bulky
- Bad readability

w/o function

```
for (int i=0; i<5; i++)
{
   cout << "Aditya Jain";
}

for (int i=0; i<5; i++)
{
   cout << "Aditya Jain";
}

for (int i=0; i<5; i++)
{
   cout << "Aditya Jain";
}
```

with function

```
void printName (string name)
{
   for (int i=0; i<5; i++)
   {
      cout << "Aditya Jain";
   }
}

printName ("Aditya Jain");
printName ("Aditya Jain");
printName ("Aditya Jain");
```

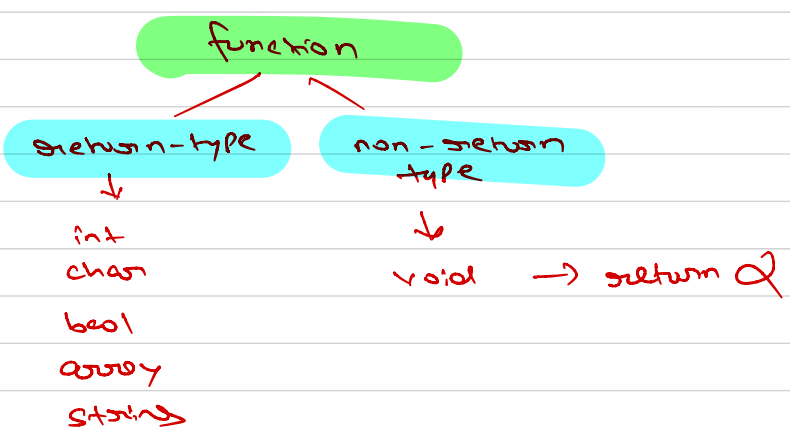
Declare

- `void printMessage();`
- `int addNum(int a, int b);`

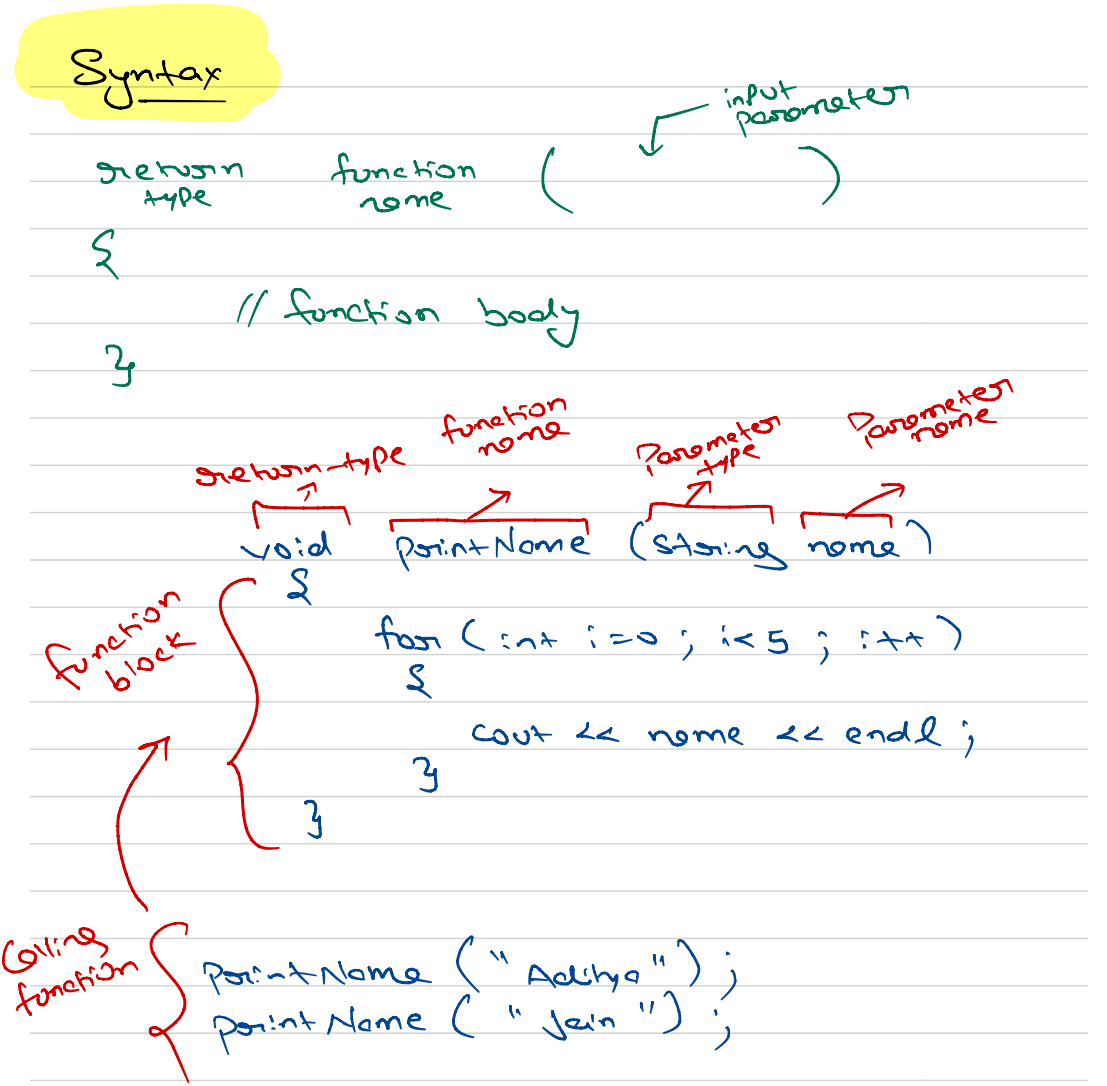
Define

-
```
int addNum(int a, int b)
{
    int sum = a + b;
    return sum;
}
```

-
```
void printName()
{
    cout << "Aditya Jain";
}
```



Syntax


input parameter
return type
function name
{
// function body
}
return-type
function name
Parameter type
Parameter name
void
PrintName
(String
name)
{
for (int i=0; i<5; i++)
{
cout << name << endl;
}
}
function block
Calling function
PrintName ("Aditya");
PrintName ("Jain");

Simple function

↳ print 10 times

↳ "Sundar ko sundari pasand hai"

```
#include <iostream>
using namespace std;

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

int main(){
    // function call
    printLine();
}
```

function does not return anything

void printLine ( )

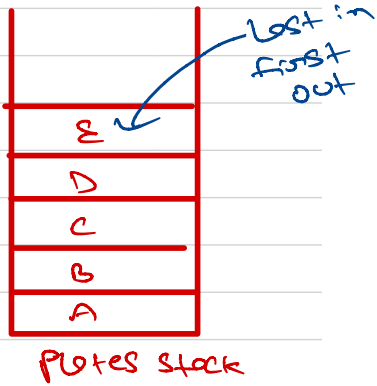
function declaration

end of function

function call

function call stack → LIFO
↳ last in first out

- ↳ function call
- ↳ local variable
- ↳ return value
- ↳ ki's func ne ki's func ko call kiya

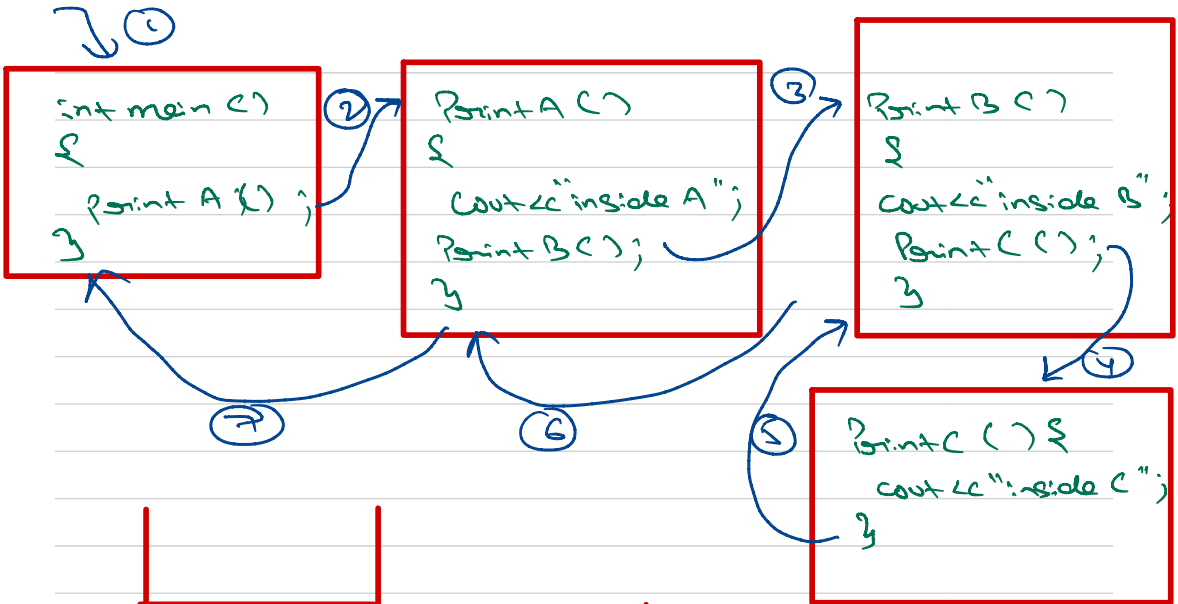



```
int main()
{
  cout << "inside main";          (1)
  PrintA();                       (2)
  cout << "Back in main";         (6)
  return 0;                       (7)
}

void PrintA()                     (3)
{
  cout << "inside A";             (4)
  cout << "going back to main";   (5)
}
```

O/P

inside main
inside A
going back to main
back in main



STACK

o/p

inside A
inside B
inside C

Void with return