Function

Monday, 4 September 2023 9:06 PM

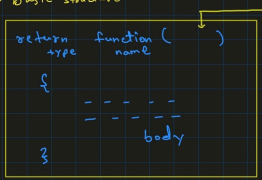
→ block of code which is used to perform a particular task
→ used for code reusability or to make the program small, less bulky
→ to increase readability

→ Every fn. has a return type :
  → void
  → int
  → string

→ A fn. take arguments as parameter

→ OS runs the program and allots memory to it.

→ basic structure



→ void fn. does not return anything

Ex →

```
void printline ()          // fn. declaration
{
    for (int i=0; i<10;i++)
    {
        cout << "Sanskar's Notes" << endl;
    }
}
int main(){
    printline();           // fn. call
    return 0;
}
```

(1) (2) (3) flow

// fn. defination

this shows that program is successfully executed and also to free memory after execution.

→ Function
- no return (void)
- return (any datatype which is used to declare fn.)

→ declaration:
Ex → void PrintMessage();
int addNum();

→ defination: int addNum (int a, int b)
{
int sum = a+b;
return sum;
}

→ call: addNum();

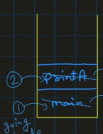
Note:
Jaha bhi fn. call karna hain uske upar hi fn. defination hona hain except jab humne fn. declare kar rakha ho fn. call ke upar.

Ex →
```
//fn. declaration
printA() {
    cout << "I am inside A";
}                          //fn. defination

int main() {
    cout << "Hi:";
    printA();              //fn. call
}
```
(2) (3) (1)

→ Function Call Stack

Ex →
```
void printA() {
    cout << "Inside A";
    cout << "going back to main";
}
int main() {
    cout << "inside main";
    printA();
    cout << "Back in main";
    return 0;
}
```



Q) stack?
→ plates in shadies

→ plate A went first
→ but will get out last
and plate E went last but will get out first
this's why
It is LIFO

Note: fn. body end or it returned then we remove that fn. from fn. call stack

Questions

1. Write a fn. to print sum of 3 no.

>> // no return type

```
void sum (int a, int b, int c){
        cout << "Sum is" << a+b+c;
        }
```

>> // now return sum of 3 no.

```
int print_sum(int a, int b, int c){
        int answer = a+b+c;
        return answer;
        }
        int main () {
                int sum;
                sum = print_sum(1,2,3);   //needs to store the return value thats why created sum variable
                cout << "Sum is " << sum;
                return 0;
                }
```

Output

>> Sum is 6

→ Can we use return keyword in void

Ex→

```
void messageprint (){
        cout << "Msg1" << endl;
        return;                    //this will return the flow to the point from where it is called.
        cout << "Msg2" << endl;
int main (){
        messageprint();
        }
```

Output

>> Msg1

Note: It will not give error if we use return in void datatype until and unless we passed a value.

Questions (Class Work)

1. Find maximum of 3 no. (a, b, c)
2. Counting from 1 to N
3. Check prime no. or not
4. Check even or odd
5. Sum of all no. upto 1 to N
6. Sum of all even no. upto 1 to N
7.

H.W

- → Fn. to find area of circle
- → fn. to find factorial of a no.
- → print all prime no. from 1 to N (V.V. Imp)
- → print all digits of an Integer
- → create a no. using digits
- → print binary repn. of a decimal no.
- → Convert km to miles
- → convert farenheit into celcium
- → Count all set bits of a number
- → check even/odd using bitwise operator

1. Max. of 3 no.

```
void print_sum (int num1, int num2, int num3)
 {
    if (num1 >= num2 && num1 >= num3)
        cout << num1 << "is greater";
    else if (num2 >= num1 && num2 >= num3)
        cout << num2 << " is greater";
    else
        cout << num3 << " is greater";
 }
```

2.
```
void printCounting (int n){
    for(int i=1; i<=n; i++){
        cout << i << endl;
    }
main(){
    printCounting (2);
    return 0;
}
```

H.W

What are the predefined or inbuilt fn. in cpp.

4.
```
void CheckEvenOdd (int num){
    if (num%2 == 0)
        cout << "Even";
    else
        cout << "Odd";
 }
```

this is not a good practice

H.W

try to do it by bitwise operator

5.
```
void findSumItoN(int num){
    int sum = 0;
    for(int i=1; i<=num; i++){
        sum = sum + i;
    }
    cout << Sum;
int main(){
    findSumItoN(5);
    return 0;
}
```

Output
» 15

6.
```
void findSumItoN(int num){
    int sum = 0;
    for(int i=1; i<=num; i++){
        if(i%2==0){
        sum = sum + i; }
    }
    cout << Sum;
int main(){
    findSumItoN(5);
    return 0;
}
```

Output
>> 6

2. Check prime or not.

A num. which is divisible by other than 1 and itself.
If a num is N, than we need to check that it is divisible by 2 to $N/2$.

```
bool (checkPrime(int n){
    for(int i=2; i<n; i++){
        if (n%i==0){
            // remainder is 0 ⇒ perfectly divisible
            // not a prime num
            return false;
        }
    }
    return true;
}
int main(){
bool prime=checkPrime(2);
    if (prime)
        cout << "Prime num";
    else
        cout << "Not a prime num";
    return 0;
}
```

// so from 2 to n-1 no num can perfectly divide N it means N is a prime num.

Output
>> 2
>> 3
>> 5
>> 7
>> 11
} all are prime