Code:

#include<stdio.h>

int main(){

int x,y,z;

x = 10; y = 20;

z = x+y;

printf("Total is %d\n", z);

printf("Cool\n");printf("Cool\n");printf("Cool\n");

x = 100; y = 200;

z = x+y;

printf("Total is %d\n", z);

printf("Cool\n");printf("Cool\n");printf("Cool\n");

x = 1000; y = 2000;

z = x+y;

printf("Total is %d\n", z);

return 0;

}

Same type of code repeating again and again but there is some code in between then we can no use looping here

Solution: we can create separate function for them

#include<stdio.h>

int main(){

int x,y,z;

x = 10; y = 20;

lmn();

printf("Cool\n");printf("Cool\n");printf("Cool\n");

x = 100; y = 200;

lmn()

printf("Cool\n");printf("Cool\n");printf("Cool\n");

return 0;

}

void lmn()

{

z = x+y;

printf("Total is %d\n", z);

}

Void: if the return type of the function is void it means the function does not return anything.

In above example our Approch is correct but code is wrong

As per rule, all the variables used inside the function should declare at the start of the function.

#include<stdio.h>

int main(){

int x,y,z;

x = 10; y = 20;

lmn();

printf("Cool\n");printf("Cool\n");printf("Cool\n");

x = 100; y = 200;

lmn();

printf("Cool\n");printf("Cool\n");printf("Cool\n");

return 0;

}

void lmn()

{

int x,y,z;

z = x+y;

printf("Total is %d\n", z);

}

This is again wrong.

We created x,y,z, for main and assign value

Then in function lmn we created x,y,z, for lmn but not assigned any value

So values are garbage and output will be garbage.

Lets take example of printf funcfiion

printf(“%d”, x); // here printf is function and we are passing value of x to print.

Like that we have the pass the values to the function.

lmn(x,y); // we are passing the value of x and y to lmn.

#include<stdio.h>

int main(){

int x,y;

x = 10; y = 20;

lmn(x,y);

printf("Cool\n");printf("Cool\n");printf("Cool\n");

x = 100; y = 200;

lmn(x,y);

return 0;

}

void lmn(int x, int y)

{

int z;

z = x+y;

printf("Total is %d\n", z);

}

Still one issue is there

Why we use include stdio.h to provice prototype of printf

In current code when control comes to lmn() then it do not know what is lmn

We are definfing lmn after the main method.

2 solutions

1. Declare the prototype of lmn before main
2. Define the method before main.

What to write in prototype: Explain that it will accept 2 int type values and return nothing.

Void lmn(int, int); // this line before main.

Final Code:

#include<stdio.h>

void lmn(int, int); // LMN prototype declaration

int main(){

int x,y;

x = 10; y = 20;

lmn(x,y); // lmn calling

printf("Cool\n");printf("Cool\n");printf("Cool\n");

x = 100; y = 200;

lmn(x,y);

return 0;

}

void lmn(int x, int y) // lmn defination

{

int z;

z = x+y;

printf("Total is %d\n", z);

}

Arguments Vs parameters

When we call the function we pass the Arguments

When we define the function then we write parameter which will accept the arguments value which will be pass at the time of calling the function.

I.e. : Those who receive the value are parameters.

lmn(10,20) // 10 and 20 are Arguments

Void lmn(int x, int y) // x and y are parameters.

{}

And number of arguments should match the number of parameters.