. Recursive Function (A function that calls itself).cpp

```cpp
//. Recursive Function (A function that calls itself)
#include <iostream>
using namespace std;

// Recursive function to calculate factorial
int factorial(int n) {
    if (n <= 1) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}

int main() {
    int num = 5;

    // Function call
    cout << "Factorial of " << num << " is: " << factorial(num) << endl;

    return 0;
}

```

F:\PSS sub\theraory pr

```
Factorial of 5 is: 120

--------------------------------
Process exited after 1.942 seconds with retu
rn value 0
Press any key to continue . . .
```

Function Overloading (Multiple Functions with the Same Name but Different Parameters),cpp.cpp

```cpp
//Function Overloading (Multiple Functions with the Same Name but Different Parameters)
#include <iostream>
using namespace std;

// Function to add two integers
int add(int a, int b) {
    return a + b;
}

// Overloaded function to add two doubles
double add(double a, double b) {
    return a + b;
}

int main() {
    // Calling the integer version
    cout << "Sum of integers: " << add(5, 10) << endl;

    // Calling the double version
    cout << "Sum of doubles: " << add(2.5, 3.5) << endl;

    return 0;
}
```

```
F:\PSS sub\theraory pr
Sum of integers: 15
Sum of doubles: 6

--------------------------------
Process exited after 1.172 seconds with retu
rn value 0
Press any key to continue . . .
```

Function Returning a Pointer.cpp

```cpp
// Function that returns a pointer to an integer
int* getPointer(int &a) {
    return &a;
}

int main() {
    int x = 42;

    // Function call to get the pointer
    int *ptr = getPointer(x);

    cout << "Value of x: " << x << endl;
    cout << "Address of x: " << *ptr << endl;
    cout << "Value at ptr: " << *ptr << endl;

    return 0;
}
```

```
F:\PSS sub\theraory pr
Value of x: 42
Address of x: 42
Value at ptr: 42

--------------------------------
Process exited after 0.9851 seconds with ret
urn value 0
Press any key to continue . . .
```

Function with Default Parameters,cpp.cpp

```cpp
//Function with Default Parameters
#include <iostream>
using namespace std;

// Function definition with a default parameter
void displayInfo(string name, int age = 18) {
    cout << "Name: " << name << ", Age: " << age << endl;
}

int main() {
    // Function calls with and without the second argument
    displayInfo("yash");
    displayInfo("dhananjay", 25);
    return 0;
}
```

void displayInfo (string name, int age = 18)

F:\PSS sub\theraory pr

```
Name: yash, Age: 18
Name: dhananjay, Age: 25

--------------------------------
Process exited after 1.093 seconds with retu
rn value 0
Press any key to continue . . .
```

Function with No Parameters and No Return Value.cpp

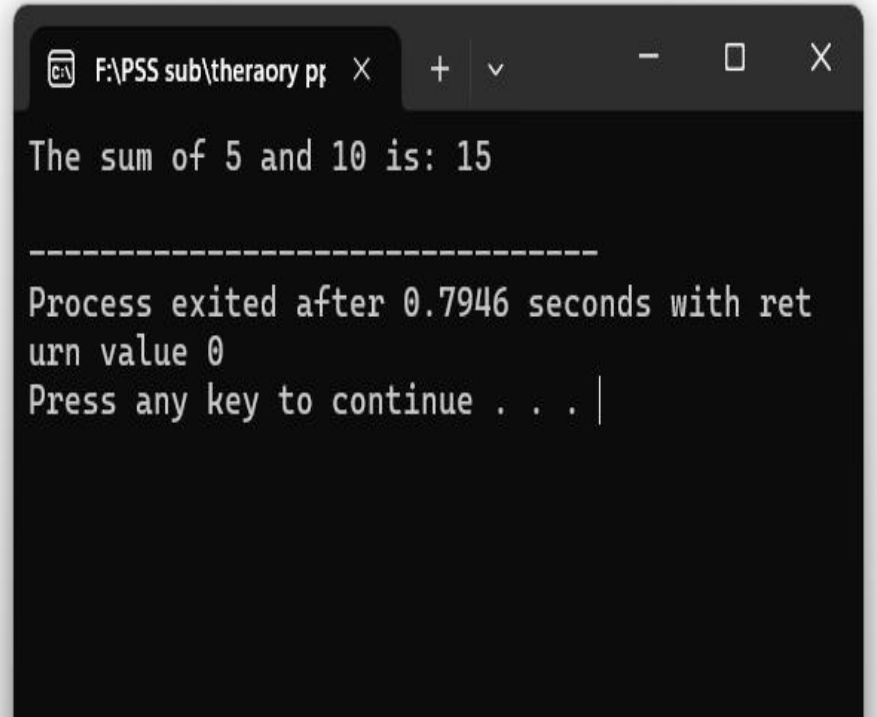
```cpp
//Function with No Parameters and No Return Value
#include <iostream>
using namespace std;

// Function definition with no parameters and no return value
void greet() {
    cout << "Hello, welcome to yash house" << endl;
}

int main() {
    // Function call
    greet();
    return 0;
}
```

F:\PSS sub\theraory pr

```
Hello, welcome to yash house

--------------------------------
Process exited after 0.924 seconds with
rn value 0
Press any key to continue . . .
```

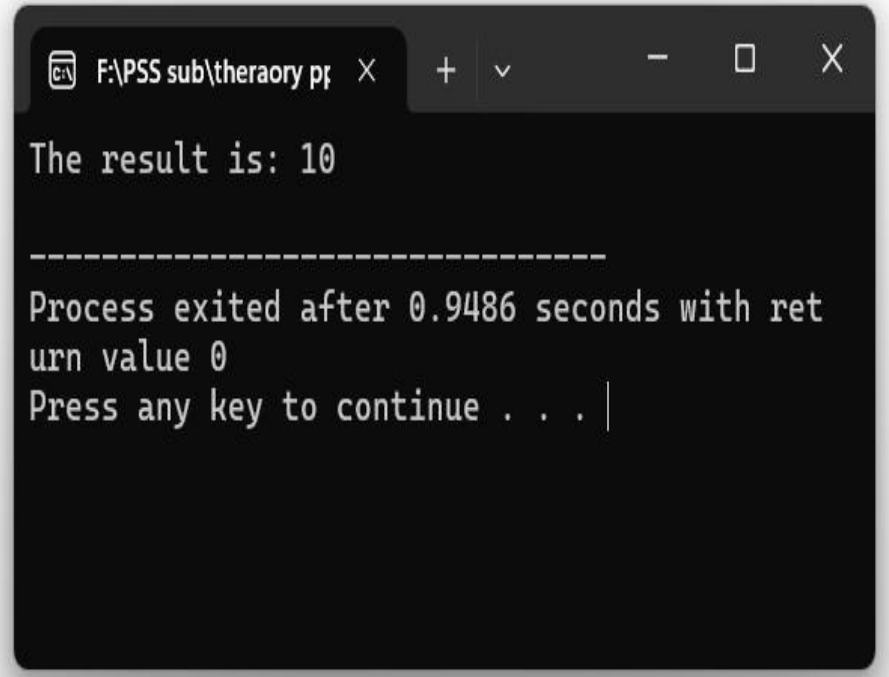
Function with Parameters and No Return Value.cpp

```cpp
//Function with Parameters and No Return Value
#include <iostream>
using namespace std;

// Function definition that takes two parameters but has no return value
void printSum(int a, int b) {
    cout << "The sum of " << a << " and " << b << " is: " << a + b << endl;
}

int main() {
    // Function call with arguments
    printSum(5, 10);
    return 0;
}
```



Function with Parameters and Return Value.cpp

```cpp
//Function with Parameters and Return Value
#include <iostream>
using namespace std;

// Function definition that takes two parameters and returns their sum
int add(int a, int b) {
    return a + b;
}

int main() {
    // Function call and storing the returned value
    int result = add(3, 7);

    cout << "The result is: " << result << endl;
    return 0;
}
```



Function with Return Type void and Reference Parameters.cpp

```cpp
//Function with Return Type void and Reference Parameters
#include <iostream>
using namespace std;

// Function that swaps two integers using reference parameters
void swap(int &a, int &b) {
    int temp = a;
    a = b;
    b = temp;
}

int main() {
    int x = 10, y = 20;

    // Before swapping
    cout << "Before swap: x = " << x << ", y = " << y << endl;

    // Function call to swap
    swap(x, y);

    // After swapping
    cout << "After swap: x = " << x << ", y = " << y << endl;

    return 0;
}

```

F:\PSS sub\theraory pr

```
Before swap: x = 10, y = 20
After swap: x = 20, y = 10

--------------------------------
Process exited after 1.284 seconds with retu
rn value 0
Press any key to continue . . .
```