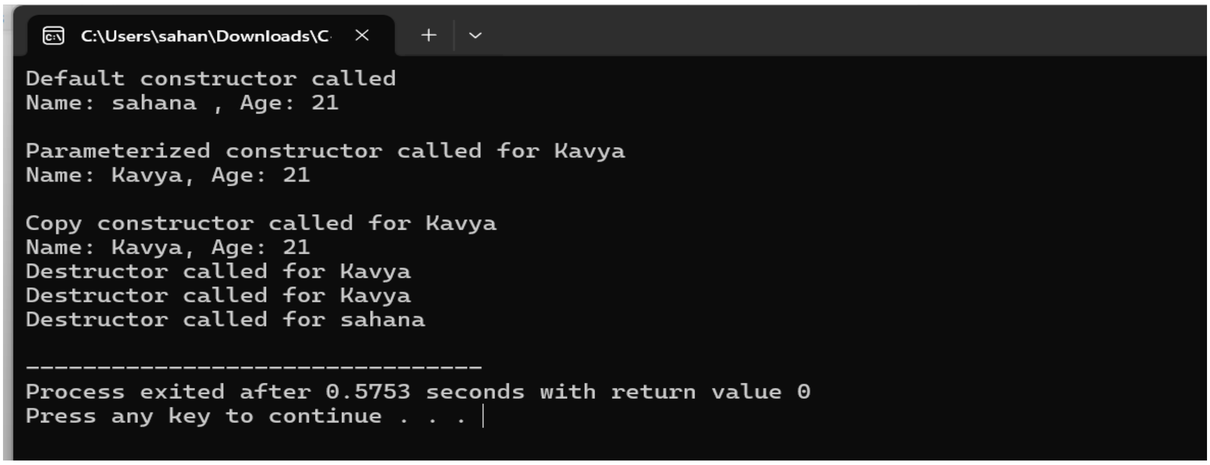
Constructors & Destructors create multiple objects with different constructors

```
#include <iostream>

using namespace std;


class Student

{

   string name;

   int age;

public:

   // Default constructor

   Student() {

      name = "sahana ";

      age = 21;

      cout << "Default constructor called" << endl;

   }

   // Parameterized constructor

   Student(string n, int a) {

      name = n;

      age = a;

      cout << "Parameterized constructor called for " << name << endl;

   }

   // Copy constructor

   Student(const Student &obj) {

      name = obj.name;

      age = obj.age;

      cout << "Copy constructor called for " << name << endl;

   }
```

```
    // Destructor
    ~Student() {
        cout << "Destructor called for " << name << endl;
    }

    void display() {
        cout << "Name: " << name << ", Age: " << age << endl;
    }
};

int main() {
    Student s1; // default constructor
    s1.display();
    cout << endl;
    Student s2("Kavya", 21); // parameterized
    s2.display();
    cout << endl;
    Student s3 = s2; // copy constructor
    s3.display();
    return 0;
}
```



```
Default constructor called
Name: sahana , Age: 21

Parameterized constructor called for Kavya
Name: Kavya, Age: 21

Copy constructor called for Kavya
Name: Kavya, Age: 21
Destructor called for Kavya
Destructor called for Kavya
Destructor called for sahana

--------------------------------
Process exited after 0.5753 seconds with return value 0
Press any key to continue . . .
```