

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN
OBJECT = CALLING THE MEMBER FUNCTION

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()  
{
```

```
    ComplexNumber c;
```

```
    cout<< "Hello there" <<endl;
```

```
    c.print();
```

```
    cout<<endl;
```

```
    c.setMemberVariables(3.14,5.3);
```

```
    cout<<endl;
```

```
    c.print();
```

```
    cout<<"Okey-dokey! All done!"<<endl;
```

```
}
```

WE ALREADY INSTANTIATED THE OBJECT IN THE PREVIOUS EXAMPLE

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()
{
    ComplexNumber c;
    cout<< "Hello there" <<endl;
    c.print();
    cout<<endl;
    c.setMemberVariables(3.14,5.3);
    cout<<endl;
    c.print();
    cout<<"Okey-dokey! All done!"<<endl;
}
```

**NOW SIMPLY CALL ITS
MEMBER FUNCTIONS!**

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()
{
    ComplexNumber c;
    cout<< "Hello there" <<endl;
    c.print();
    cout<<endl;
    c.setMemberVariables(3.14,5.3);
    cout<<endl;
    c.print();
    cout<<"Okey-dokey! All done!"<<endl;
}
```

**USE THE DOT OPERATOR AS
WITH STRUCTS**

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()
{
    ComplexNumber c;
    cout<< "Hello there" <<endl;
    c.print();
    cout<<endl;
    c.setMemberVariables(3.14,5.3);
    cout<<endl;
    c.print();
    cout<<"Okey-dokey! All done!"<<endl;
}
```

PASS IN ARGUMENTS AS NEEDED (AS YOU WOULD WITH ANY FUNCTION)

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()
{
    ComplexNumber c;
    cout<< "Hello there" <<endl;
    c.print();
    cout<<endl;
    c.setMemberVariables(3.14,5.3);
    cout<<endl;
    c.print();
    cout<<"Okey-dokey! All done!"<<endl;
}
```

REMEMBER THAT EACH OBJECT HAS ITS OWN COPY OF THE MEMBER VARIABLES -

```
class ComplexNumber
{
private:
    float realPart;
    float complexPart;
public:
    ComplexNumber()
    {
        cout << "No arg-constructor called" << endl;
    }
    void setMemberVariables(double r,double c)
    {
        realPart = r;
        complexPart = c;
    }
    float getRealPart()
    {
        return realPart;
    }

    float getComplexPart()
    {
        return complexPart;
    }
    void print()
    {
        cout<<"real = " << realPart << " complex = " << complexPa
    }
}
```


EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()
{
    ComplexNumber c;
    cout<< "Hello there" <<endl;
    c.print();
    cout<<endl;
    c.setMemberVariables(3.14,5.3);
    cout<<endl;
    c.print();
    cout<<"Okey-dokey! All done!"<<endl;
}
```

REMEMBER THAT EACH OBJECT HAS
ITS OWN COPY OF THE MEMBER
VARIABLES -

```
class ComplexNumber
{
private:
    float realPart;
    float complexPart;
public:
    ComplexNumber()
    {
        cout << "No arg-constructor called" << endl;
    }
    void setMemberVariables(double r,double c)
    {
        realPart = r;
        complexPart = c;
    }
    float getRealPart()
    {
        return realPart;
    }
    float getComplexPart()
    {
        return complexPart;
    }
    void print()
    {
        cout<<"real = " << realPart << " complex = " << complexPa
    }
}
```


EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()
{
    ComplexNumber c;
    cout<< "Hello there" <<endl;
    c.print();
    cout<<endl;
    c.setMemberVariables(3.14,5.3);
    cout<<endl;
    c.print();
    cout<<"Okey-dokey! All done!"<<endl;
}
```

**THE MEMBER FUNCTIONS WILL
OPERATE ON THOSE SPECIFIC COPIES
(BELONGING TO THAT OBJECT)**

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out  
No arg-constructor called  
Hello there
```

```
real = 7.29249e+19 complex = 4.59163e-41
```

```
real = 3.14 complex = 5.30key-dokey! All done!
```

THE OUTPUT PROVES THIS!

```
ComplexNumber c;  
cout<< "Hello there" <<endl;  
c.print();  
cout<<endl;  
c.setMemberVariables(3.14,5.3);  
cout<<endl;  
c.print();  
cout<<"key-dokey! All done!"<<endl;  
}
```


EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out
```

```
No arg-constructor called
```

```
Hello there
```

```
real = 7.29249e+19 complex = 4.59163e-41
```

```
real = 3.14 complex = 5.30key-dokey! All done!
```

```
ComplexNumber c;
```

```
cout<< "Hello there" <<endl;
```

```
c.print();
```

```
cout<<endl;
```

```
c.setMemberVariables(3.14,5.3);
```

```
cout<<endl;
```

```
c.print();
```

```
cout<<"key-dokey! All done!"<<endl;
```

```
}
```

THE OUTPUT PROVES THIS!

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out
```

```
No arg-constructor called
```

```
Hello there
```

```
real = 7.29249e+19 complex = 4.59163e-41
```

```
real = 3.14 complex = 5.30key-dokey!{
```

```
ComplexNumber c;
```

```
cout<< "Hello there" <<endl;
```

```
c.print();
```

```
cout<<endl;
```

```
c.setMemberVariables(3.14,5.3);
```

```
cout<<endl;
```

```
c.print();
```

```
cout<<"0key-dokey! All done!"<<endl;
```

```
}
```

THE OUTPUT PROVES THIS!

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out
```

```
No arg-constructor called
```

```
Hello there
```

```
real = 7.29249e+19 complex = 4.59163e-41
```

```
int main()  
real = 3.14 complex = 5.30key-dokey!{
```

```
ComplexNumber c;
```

```
cout<< "Hello there" <<endl;
```

```
c.print();
```

```
cout<<endl;
```

```
c.setMemberVariables(3.14,5.3);
```

```
cout<<endl;
```

```
c.print();
```

```
cout<<"Okey-dokey! All done!"<<endl;
```

```
}
```

THE OUTPUT PROVES THIS!

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out  
No arg-constructor called  
Hello there  
real = 7.29249e+19 complex = 4.59163e-41  
int main()  
real = 3.14 complex = 5.30key-dokey!{
```

THE OUTPUT PROVES THIS!

```
ComplexNumber c;  
cout<< "Hello there" <<endl;  
c.print();  
cout<<endl;  
c.setMemberVariables(3.14,5.3);  
cout<<endl;  
c.print();  
cout<<"0key-dokey! All done!"<<endl;  
}
```


EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out
```

```
No arg-constructor called
```

```
Hello there
```

```
real = 7.29249e+19 complex = 4.59163e-41
```

```
real = 3.14 complex = 5.30key-dokey!
```

THE OUTPUT PROVES THIS!

```
int main()  
{  
    ComplexNumber c;  
    cout<< "Hello there" <<endl;  
    c.print();  
    cout<<endl;  
    c.setMemberVariables(3.14,5.3);  
    cout<<endl;  
    c.print();  
    cout<<"Okey-dokey! All done!"<<endl;  
}
```

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out
```

```
No arg-constructor called
```

```
Hello there
```

```
real = 7.29249e+19 complex = 4.59163e-41
```

```
real = 3.14 complex = 5.30key-dokey!
```

THE OUTPUT PROVES THIS!

```
int main()  
{  
    ComplexNumber c;  
    cout<< "Hello there" <<endl;  
    c.print();  
    cout<<endl;  
    c.setMemberVariables(3.14,5.3);  
    cout<<endl;  
    c.print();  
    cout<<"0key-dokey! All done!"<<endl;  
}
```


EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ ./a.out
```

```
No arg-constructor called
```

```
Hello there
```

```
real = 7.29249e+19 complex = 4.59163e-41
```

```
real = 3.14 complex = 5.30key-dokey!
```

THE OUTPUT PROVES THIS!

```
int main()
{
    ComplexNumber c;
    cout<< "Hello there" <<endl;
    c.print();
    cout<<endl;
    c.setMemberVariables(3.14,5.3);
    cout<<endl;
    c.print();
    cout<<"0key-dokey! All done!"<<endl;
}
```


EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

REMEMBER THAT EACH OBJECT HAS ITS OWN COPY OF THE MEMBER VARIABLES -

C1 AND C2 ARE DIFFERENT INSTANCES AND THEY HAVE THEIR OWN COPY OF THE VARIABLES

CHANGES TO C1 DOES NOT AFFECT C2!

```
int main()
{
    ComplexNumber c1, c2;
    cout << "C1 holds" << endl;
    c1.print();
    cout << endl << "C2 holds" << endl;
    c2.print();
    cout << endl;

    // Update C1
    c1.setMemberVariables(3.14,5.3);
    cout << endl;
    // Note that C1 has been updated with the new value
    c1.print();

    cout << endl;
    // C2 remains unchanged
    c2.print();
    cout << endl;
    cout << "Okey-dokey! All done!" << endl;
}
```

```
class ComplexNumber
{
private:
    float realPart;
    float complexPart;
public:
    ComplexNumber()
    {
        cout << "No arg-constructor called" << endl;
    }
    void setMemberVariables(double r,double c)
    {
        realPart = r;
        complexPart = c;
    }
}
```

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

CHANGES TO C1 DOES NOT AFFECT C2!

```
int main()
{
    ComplexNumber c1, c2;
    cout << "C1 holds" << endl;
    c1.print();
    cout << endl << "C2 holds" << endl;
    c2.print();
    cout << endl;

    // Update C1
    c1.setMemberVariables(3.14,5.3);
    cout << endl;
    // Note that C1 has been updated with the new values
    c1.print();

    cout << endl;
    // C2 remains unchanged
    c2.print();
    cout << endl;
    cout << "Okey-dokey! All done!" << endl;
}
```

No arg-constructor called
No arg-constructor called

C1 holds
real = 1.71744e+22 complex = 4.59163e-41
C2 holds
real = 2.78242e-35 complex = 1.4013e-45

real = 3.14 complex = 5.3
real = 2.78242e-35 complex = 1.4013e-45
Okey-dokey! All done!

2 OBJECT INSTANTIATED, THE DEFAULT
CONSTRUCTOR IS CALLED TWICE

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

CHANGES TO C1 DOES NOT AFFECT C2!

```
int main()
{
    ComplexNumber c1, c2;
    cout << "C1 holds" << endl;
    c1.print();
    cout << endl << "C2 holds" << endl;
    c2.print();
    cout << endl;

    // Update C1
    c1.setMemberVariables(3.14,5.3);
    cout << endl;
    // Note that C1 has been updated with the new values
    c1.print();

    cout << endl;
    // C2 remains unchanged
    c2.print();
    cout << endl;
    cout << "Okey-dokey! All done!" << endl;
}
```

```
No arg-constructor called
No arg-constructor called
C1 holds
real = 1.71744e+22 complex = 4.59163e-41
C2 holds
real = 2.78242e-35 complex = 1.4013e-45

real = 3.14 complex = 5.3
real = 2.78242e-35 complex = 1.4013e-45
Okey-dokey! All done!
```

C1 HOLDS SOME VALUE

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

CHANGES TO C1 DOES NOT AFFECT C2!

```
int main()
{
    ComplexNumber c1, c2;
    cout << "C1 holds" << endl;
    c1.print();
    cout << endl << "C2 holds" << endl;
    c2.print();
    cout << endl;

    // Update C1
    c1.setMemberVariables(3.14,5.3);
    cout << endl;
    // Note that C1 has been updated with the new values
    c1.print();

    cout << endl;
    // C2 remains unchanged
    c2.print();
    cout << endl;
    cout << "Okey-dokey! All done!" << endl;
}
```

```
No arg-constructor called
No arg-constructor called
C1 holds
real = 1.71744e+22 complex = 4.59163e-41
C2 holds
real = 2.78242e-35 complex = 1.4013e-45

real = 3.14 complex = 5.3
real = 2.78242e-35 complex = 1.4013e-45
Okey-dokey! All done!
```

C1 HOLDS SOME VALUE

...WHICH IS COMPLETELY DIFFERENT FROM C2

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

```
int main()
{
    ComplexNumber c1, c2;
    cout << "C1 holds" << endl;
    c1.print();
    cout << endl << "C2 holds" << endl;
    c2.print();
    cout << endl;

    // Update C1
    c1.setMemberVariables(3.14, 5.3);
    cout << endl;
    // Note that C1 has been updated with the new values
    c1.print();

    cout << endl;
    // C2 remains unchanged
    c2.print();
    cout << endl;
    cout << "Okey-dokey! All done!" << endl;
}
```

CHANGES TO C1 DOES NOT
AFFECT C2!

No arg-constructor called
No arg-constructor called
C1 holds
real = 1.71744e+22 complex = 4.59163e-41
C2 holds
real = 2.78242e-35 complex = 1.4013e-45
real = 3.14 complex = 5.3
real = 2.78242e-35 complex = 1.4013e-45
Okey-dokey! All done!

C1'S VALUE IS EXPLICITLY
SET - IT HAS BEEN
UPDATED

EXAMPLE 3: INVOKE MEMBER FUNCTION OF AN OBJECT

INVOKING A MEMBER FUNCTION OF AN OBJECT = CALLING THE MEMBER FUNCTION

CHANGES TO C1 DOES NOT AFFECT C2!

```
int main()
{
    ComplexNumber c1, c2;
    cout << "C1 holds" << endl;
    c1.print();
    cout << endl << "C2 holds" << endl;
    c2.print();
    cout << endl;

    // Update C1
    c1.setMemberVariables(3.14,5.3);
    cout << endl;
    // Note that C1 has been updated with the new values
    c1.print();

    cout << endl;
    // C2 remains unchanged
    c2.print();
    cout << endl;
    cout << "Okey-dokey! All done!" << endl;
}
```

```
No arg-constructor called
No arg-constructor called
C1 holds
real = 1.71744e+22 complex = 4.59163e-41
C2 holds
real = 2.78242e-35 complex = 1.4013e-45

real = 3.14 complex = 5.3
real = 2.78242e-35 complex = 1.4013e-45
Okey-dokey! All done!
```

C2 DOES NOT CHANGE!