EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS
= CREATING A VARIABLE OF THAT CLASS

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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;

}
```

CREATE A VARIABLE OF THIS CLASS, JUST LIKE YOU WOULD OF ANY OTHER TYPE!

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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 NOW HAVE AN OBJECT, NAMED c...

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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 NOW HAVE AN OBJECT, NAMED c...

OF THE CLASS NAMED

ComplexNumber

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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 NOW HAVE AN OBJECT, NAMED c... OF THE CLASS NAMED ComplexNumber

WE INSTANTIATED AN OBJECT OF OUR CLASS!

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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;

}
```

WHEN THIS LINE IS EXECUTED, C++ WILL AUTOMATICALLY CALL THE NO-ARGUMENT CONSTRUCTOR OF THE CLASS

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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 = " << complexPart;
  }
};
```

NO ARGUMENTS

WHEN THIS LINE IS EXECUTED, C++ WILL AUTOMATICALLY CALL THE NO-ARGUMENT CONSTRUCTOR OF THE CLASS

AS YOU CAN SEE FROM THIS, THE C++ COMPILER RELIES ON EVERY CLASS HAVING A CONSTRUCTOR THAT TAKES IN NO ARGUMENTS

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

AS YOU CAN SEE FROM THIS, THE C++ COMPILER RELIES ON EVERY CLASS HAVING A CONSTRUCTOR THAT TAKES IN NO ARGUMENTS

"DEFAULT CONSTRUCTOR"

IF YOU FORGET TO ADD A NO-ARG CONSTRUCTOR TO YOUR CLASS, C++ WILL ADD ONE FOR YOU!

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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 do

}
```

WHEN THIS LINE IS EXECUTED, C++ WILL AUTOMATICALLY CALL THE NO-ARGUMENT CONSTRUCTOR OF THE CLASS

"DEFAULT CONSTRUCTOR"

```
ComplexNumber()
{
   cout << "No arg-constructor called" << endl;
}
```

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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 do

}
```

WHEN THIS LINE IS EXECUTED, C++ WILL AUTOMATICALLY CALL THE NO-ARGUMENT CONSTRUCTOR OF THE CLASS

"DEFAULT CONSTRUCTOR"

PRINTS OUT THIS MESSAGE..

```
ComplexNumber()
{
   cout << "No arg-constructor called" << endl;
}
```

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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!
```

PRINTS OUT THIS MESSAGE..

PROVED BY RUNNING THE CODE!

WHEN THIS LINE IS EXECUTED, C++ WILL AUTOMATICALLY CALL THE NO-ARGUMENT CONSTRUCTOR OF THE CLASS

"DEFAULT CONSTRUCTOR"

EXAMPLE 2: INSTANTIATE AN OBJECT OF A REALLY SIMPLE CLASS

INSTANTIATING AN OBJECT OF A CLASS = CREATING A VARIABLE OF THAT CLASS

```
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!
```

PROVED BY RUNNING THE CODE!

WHICH WE DID AFTER COMPILING THE CODE!

```
Vitthals-MacBook-Pro:~ vitthalsrinivasan$ g++ name.cpp
```