EXAMPLE 12:

THE PLACEMENT NEW OPERATOR

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC MEMORY LOCATION (LIKE WITH `malloc`)

PLACEMENT NEW TAKES IN THE MEMORY LOCATION AS A PARAMETER

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC MEMORY LOCATION (LIKE WITH `malloc`)

PLACEMENT NEW TAKES IN THE MEMORY LOCATION AS A PARAMETER

IF YOU USE PLACEMENT NEW, YOU HAVE TO EXPLICITLY CALL THE DESTRUCTOR!

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC MEMORY LOCATION (LIKE WITH `malloc`)

PLACEMENT NEW TAKES IN THE MEMORY LOCATION AS A PARAMETER

IF YOU USE PLACEMENT NEW, YOU HAVE TO EXPLICITLY CALL THE DESTRUCTOR!

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT
ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC
MEMORY LOCATION (LIKE WITH `malloc`)

```
int main()
{
  char memoryBuffer[sizeof(ComplexNumber)];
  void *place = (void*)memoryBuffer;

  ComplexNumber * cPlacedNew = new(place) ComplexNumber(10,5);
  cout << "Printing out dynamically allocated object that used placement new" << endl;
  cPlacedNew->print();

  // Now need to explicitly call the destructor when using placement new
  cPlacedNew->~ComplexNumber();

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


}
```

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT
ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC
MEMORY LOCATION (LIKE WITH `malloc`)

FIRST SET UP A MEMORY AREA USING SIZEOF

```
int main()
{
  char memoryBuffer[sizeof(ComplexNumber)];
  void *place = (void*)memoryBuffer;

  ComplexNumber * cPlacedNew = new(place) ComplexNumber(10,5);
  cout << "Printing out dynamically allocated object that used placement new" << endl;
  cPlacedNew->print();

  // Now need to explicitly call the destructor when using placement new
  cPlacedNew->~ComplexNumber();

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


}
```

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC MEMORY LOCATION (LIKE WITH `malloc`)

FIRST SET UP A MEMORY AREA USING SIZEOF

```
int main()
{
  char memoryBuffer[sizeof(ComplexNumber)];
  void *place = (void*)memoryBuffer;

  ComplexNumber * cPlacedNew = new(place) ComplexNumber(10,5);
  cout << "Printing out dynamically allocated object that used placement new" << endl;
  cPlacedNew->print();

  // Now need to explicitly call the destructor when using placement new
  cPlacedNew->~ComplexNumber();

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


}
```

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF new, WHICH THAT ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC MEMORY LOCATION (LIKE WITH malloc)

```
int main()
{
  char memoryBuffer[sizeof(ComplexNumber)];
  void *place = (void*)memoryBuffer;

  ComplexNumber * cPlacedNew = new(place) ComplexNumber(10,5);
  cout << "Printing out dynamically allocated object that used placement new" << endl;
  cPlacedNew->print();

  // Now need to explicitly call the destructor when using placement new
  cPlacedNew->~ComplexNumber();

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

}
```

PLACEMENT NEW TAKES IN THE MEMORY LOCATION AS A PARAMETER

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC MEMORY LOCATION (LIKE WITH `malloc`)

```
int main()
{
  char memoryBuffer[sizeof(ComplexNumber)];
  void *place = (void*)memoryBuffer;

  ComplexNumber * cPlacedNew = new(place) ComplexNumber(10,5);
  cout << "Printing out dynamically allocated object that used placement new" << endl;
  cPlacedNew->print();

  // Now need to explicitly call the destructor when using placement new
  cPlacedNew->~ComplexNumber();

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


}
```

IF YOU USE PLACEMENT NEW, YOU HAVE TO EXPLICITLY CALL THE DESTRUCTOR!

EXAMPLE 12: THE PLACEMENT NEW OPERATOR

THIS IS A VARIATION OF `new`, WHICH THAT ENSURES THAT AN OBJECT RESIDES AT A SPECIFIC MEMORY LOCATION (LIKE WITH `malloc`)

IF YOU USE PLACEMENT NEW, YOU HAVE TO EXPLICITLY CALL THE DESTRUCTOR!

DON'T USE PLACEMENT `new` UNLESS YOU ABSOLUTELY HAVE TO (EG IN HARDWARE APPLICATIONS)