Printing Out Class Member Data Solutions

First attempt

- Add a print() member function to the class below which displays the value of the data members on the screen
- Create a program which calls the print() method

```
class Test {
   int i{42};
   string str;
   ...
};
```

First attempt

- What are the disadvantages of this solution?
  - cout is hard-coded
  - This function cannot be used with other streams
- Suggest an improvement
  - Use an ostream variable
- How does your suggestion address these disadvantages?
  - It allows the data to be sent to any output stream

Second attempt

- Modify the print() member function so that it takes a std::ostream as its argument and sends the data there
- Create a program which implements the print() method and calls it, passing cout as the output stream argument
- Does it matter whether the output stream is passed by reference or by value?
  - Streams cannot be passed by value as they cannot be copied
- Modify your program so that it opens a file. Call the print() member function to save the data to the file

Compatibility with Output of Built-in Types

- Explain why the code below does not work
  - cout << "Test object: " << test << endl;
  - Stream operators << are only provided for built-in and library types
  - There is no operator << that can take a Test object as its argument
- What changes would you need to make for the code to work?
- We need to provide a suitable operator <<

Nested Calls of Operator <<

- How will the code below be invoked?

```
int i{1}, j{2};
cout << i << j;
operator <<(operator <<(cout, j), i);
```

Nested Calls of Operator <<

- Explain why the overloaded << operator returns a reference to the output stream
  - The output stream is not copyable, so it can only be returned by reference
  - Returning the stream means that operator << calls can be chained

    operator <<(operator <<(cout, j), i);
  - The next call in the chain needs to modify the stream by pushing data on it, so it must be a modifiable reference

Overloaded << operator for Test

- Modify your program so that the code below compiles and runs correctly

```
cout << "Test object: " << test << endl;
ofile << "Test object : " << test << endl;
```