

NAME

Solutions

SS# \_\_\_\_\_

Create a class called message. It is to have as private a string that holds a message, an integer that holds a destination location number, another integer that holds the source location number and a flag that indicates that the message has been displayed. It is to have as public a function that displays the contents of the message, its source and destination address. This function is to set the flag to indicate the message has been displayed. The constructor is to initialize the message and its source and destination location numbers and set the flag to indicate the message has not yet been displayed. The destructor checks the flag. If the flag indicates that the message has not yet been displayed it will display a warning to this effect.

```
class message
```

```
{
```

```
private:
```

```
    char    msg[80];
```

```
    int     dest;
```

```
    int     source;
```

```
    int     displayed;
```

```
public:
```

```
    void    display();
```

```
    message(char m[80], int d, int s);
```

```
    ~message();
```

```
}
```

```
void message::display()
```

```
{
```

```
printf(" %s to %d from %d \n",  
      msg, source, dest);
```

```
displayed = TRUE;
```

```
}
```

```
message::message(char m[80], int d, int s)
```

```
{
```

```
strcpy(msg, m);
```

```
dest = d;
```

```
source = s;
```

```
displayed = FALSE;
```

```
}
```

```
message::~~message()
```

```
{
```

```
if (!displayed)
```

```
printf("message %s not displayed \n",  
      msg);
```

```
}
```