

```
class Cls : public QObject
{
```

```
    public:
```

```
        void trigger_sig_1()
```

```
        {
```

```
            // don't forget the context of this pointer
```

```
            emit sig_1();
```

```
        }
```

```
    signals:
```

```
        void sig_1();
```

```
};
```

```
// how can I emit this signal?
```

```
Cls c1; // type(c1) == Cls
```

```
Cls c2; // type(c2) == Cls
```

```
c1.trigger_sig_1(); // Cls::trigger_sig_1(&c1);
```

```
c2.trigger_sig_1(); // Cls::trigger_sig_2(&c2);
```

```
// signal emitted by a call c1.trigger_sig_1() is considered/recognized to be a different signal
```

```
// than what is raised by a call to c2.trigger_sig_2()
```

```
// although at source code level it appears that the same signal has been emitted twice
```

$C1 \longrightarrow \text{Signals: } \underline{\text{sig1}_1, \text{sig2}_1, \dots, \text{Sig}^n_1}$

$OC1, OC2, \dots, OCm. \quad | \text{ } m \text{ objects}$

$m \times n = \text{Different Signals can be emitted}$

- In order to emit a signal from sig1_1 to

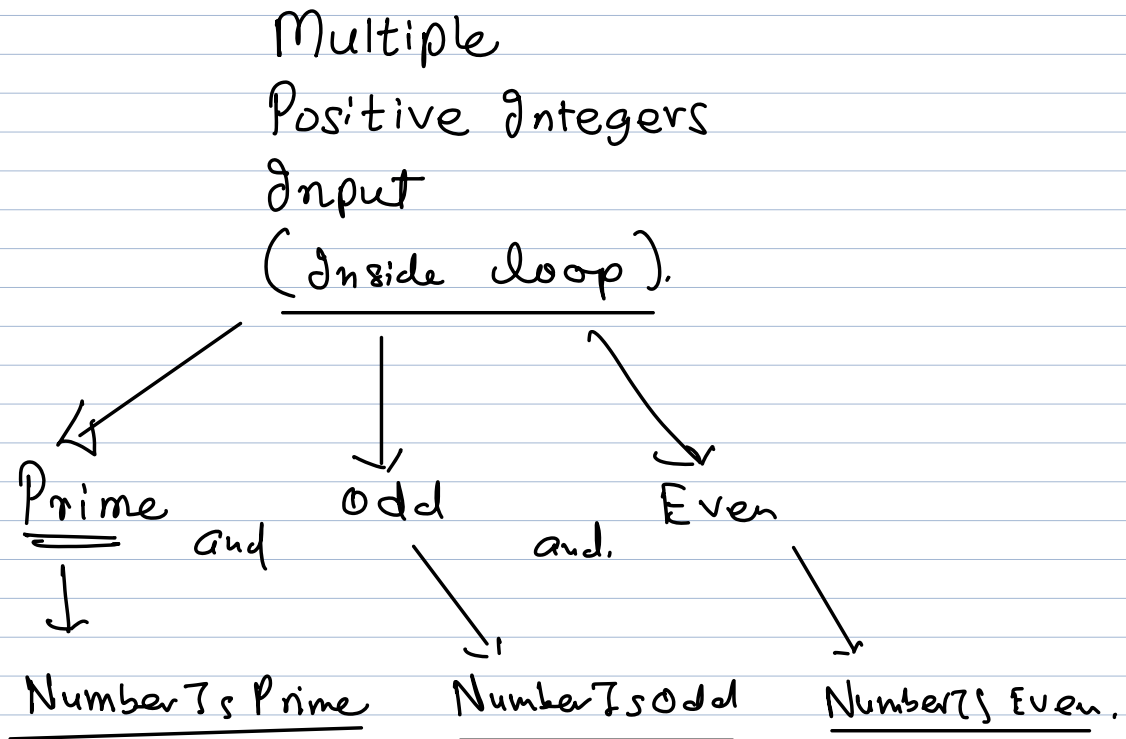
Sig^n_1 , the class $C1$ must have at least

one non-static member function containing

emit statement for that signal.

- In order to reach that non-static member

function we must use one of the objects of
C1.



Prime Number Accumulator → vec

Odd Number Accumulator → vec

Even Number Accumulator → vec

Show Your Vector

→ Show