```
#include <iostream>
using namespace std;

class Subject {
public:
  virtual void f() = 0;
};

class D2 : public Subject {
  void g();
};

class Singleton
{
private:
   Singleton();

public:
   static Singleton& instance()
   {
      static Singleton INSTANCE;
      return INSTANCE;
   }
};

class Observer
{
   protected:
   Subject *subject;
   public:
   virtual void update() = 0;
  virtual ~Observer()
  {
    delete subject;
  }

};

class BinaryObserver : public Observer
{

   public:
   BinaryObserver(Subject *subject);

   void update() override;
};

class OctalObserver : public Observer
{

   public:
   OctalObserver(Subject *subject);

   void update() override;
};

class HexaObserver : public Observer
{

   public:
   HexaObserver(Subject *subject);

   void update() override;
};


BinaryObserver::BinaryObserver(Subject *subject)
{
   this->subject = subject;
   this->subject->attach(this);
}

void BinaryObserver::update()
{
   std::wcout << L"Binary String: " << Integer::toBinaryString(subject-
>getState()) << std::endl;
}

OctalObserver::OctalObserver(Subject *subject)
{
   this->subject = subject;
   this->subject->attach(this);
}

void OctalObserver::update()
{
  std::wcout << L"Octal String: " << Integer::toOctalString(subject->getState())
<< std::endl;
}
```

```
HexaObserver::HexaObserver(Subject *subject)
{
   this->subject = subject;
   this->subject->attach(this);
}

void HexaObserver::update()
{
   std::wcout << L"Hex String: " << Integer::toHexString(subject->getState())-
>toUpperCase() << std::endl;
}

int main() {
  D2 d;
}
```