Serialization Framework

Write serialization framework with virtual inheritance objects

- The framework needs to be able to store and restore any list of instances
- Objects do not have any knowledge about the file format
- Write a generic solution that can be used for any class that derived from Base without any additional changes to the serialization framework

You should:

1. Implement OutArcive and InArchive.
2. Write reader/writer functions of Base and its derivatives.

- No limitations regarding any additional helper class.

sample:

```
class Base
{
protected::
        std::string name
};

class Point : public Base
{
private:
        float x;
        float y;
};

class Circle : public Point
{
private:

        int r;
};

int main ()
{
        std::vector<std::shared_ptr<Base>> objects;
        objects.push_back(std::make_shared<Point>(1, 2));
        objects.push_back(std::make_shared<Point>(50, 12));
        objects.push_back(std::make_shared<Circle>(0, 0, 10));

        {
                OutArchive write_archive(“out.dat”);
                write_archive.write(objects);
        }

        std::vector<std::shared_ptr<Base>> loaded_objects;

        InArchive read_archive(“out.dat”);
        read_archive.read(loaded_objects);
        // loaded_objects should contain the same objects as in objects
}
```