Начала объектно-ориентированного программирования

Шаг 1

```
#include <iostream>

using namespace std;

class Double
{
  double number;

  public:
  Double(double d = 0.0) : number(d) {}
  Double add(Double, Double);
  friend ostream& operator<<(ostream&, Double);
};

Double Double::add(Double a, Double b)
{
  Double temporary;
  temporary.number = a.number + b.number;
  return temporary;
}

ostream& operator<<(ostream& stream, Double d)
{
  return stream << d.number;
}

int main()
{
  Double a(1.2), b(2.3), c;
  cout << "a = " << a << endl;                 // a = 1.2
  cout << "b = " << b << endl;                 // b = 2.3
  cout << "c = " << c << endl;                 // c = 0
  cout << "a + b = " << a.add(a, b) << endl;  // a + b = 3.5
  cout << "a + b = " << b.add(a, b) << endl;  // a + b = 3.5
  cout << "a + b = " << c.add(a, b) << endl;  // a + b = 3.5

  return 0;
}
```

Шаг 1. Версия 1

```
#include <iostream>

using namespace std;

class Double
{
  double number;

  public:
  Double(double d = 0.0) : number(d) {}
  Double add(Double, Double);
  void print(Double);
};

Double Double::add(Double a, Double b)
{
  Double temporary;
  temporary.number = a.number + b.number;
  return temporary;
}

void Double::print(Double d)
{
  cout << d.number << endl;
}

int main()
{
  Double a(1.2), b(2.3), c;
  cout << "a = ";
  a.print(a);                // a = 1.2
  cout << "b = ";
  b.print(b);                // b = 2.3
  cout << "c = ";
  c.print(c);                // c = 0
  cout << "a + b = ";
  a.print(a.add(a, b));      // a + b = 3.5

  return 0;
}
```

Шаг 1. Версия 2

```
#include <iostream>

class Double
{
  public:
  double number;

  Double(double d = 0.0) : number(d) {}
  Double add(Double, Double);
};

Double Double::add(Double a, Double b)
{
  Double temporary;
  temporary.number = a.number + b.number;
  return temporary;
}

int main()
{
  using namespace std;

  Double a(1.2), b(2.3), c;
  cout << "a = " << a.number << endl;  // a = 1.2
  cout << "b = " << b.number << endl;  // b = 2.3
  cout << "c = " << c.number << endl;  // c = 0
  c = a.add(a, b);
  cout << "c = " << c.number << endl;  // c = 3.5
  a.number = 3.4;
  b.number = 4.5;
  cout << "a = " << a.number << endl;  // a = 3.4
  cout << "b = " << b.number << endl;  // b = 4.5
  c = a.add(a, b);
  cout << "c = " << c.number << endl;  // c = 7.9

  return 0;
}
```

Шаг 2

```
#include <iostream>

class Double
{
  public:
  double number;
};

int main()
{
  using namespace std;

  Double a, b;
  a.number = 1.2;
  b.number = 2.3;
  cout << "a = " << a.number << endl;                // a = 1.2
  cout << "b = " << b.number << endl;                // b = 2.3
  cout << "a + b = " << a.number + b.number << endl; // a + b = 3.5

  return 0;
}
```

Шаг 2. Версия 1

```
#include <iostream>

using namespace std;

class Double
{
  public:
  double number;

  Double()
  {
    number = 0.0;
    cout << "It's my default constructor" << endl;
  }
};

int main()
{
  Double a;
  cout << "a = " << a.number << endl;                // a = 0
  Double b;
  cout << "b = " << b.number << endl;                // b = 0
  a.number = 1.2;
  b.number = 2.3;
  cout << "a = " << a.number << endl;                // a = 1.2
  cout << "b = " << b.number << endl;                // b = 2.3
  cout << "a + b = " << a.number + b.number << endl; // a + b = 3.5

  return 0;
}
```

Шаг 3

```
#include <iostream>

class Double
{
  public:
  double number;
};

int main()
{
  using namespace std;

  Double a;
  a.number = 1.2;
  cout << "a = " << a.number << endl;      // a = 1.2
  Double b = a;
  cout << "b = " << b.number << endl;      // b = 1.2
  Double c(b);
  cout << "c = " << c.number << endl;      // c = 1.2

  return 0;
}
```

Шаг 3. Версия 1

```
#include <iostream>

using namespace std;

class Double
{
  public:
  double number;

  Double()
  {
    number = 0.0;
    cout << "It's my default constructor" << endl;
  }

  Double(const Double& object)
  {
    number = object.number;
    cout << "It's my copy constructor" << endl;
  }
};

int main()
{
  Double a;
  cout << "a = " << a.number << endl;      // a = 0
  a.number = 1.2;
  cout << "a = " << a.number << endl;      // a = 1.2
  Double b = a;
  cout << "b = " << b.number << endl;      // b = 1.2
  Double c(b);
  cout << "c = " << c.number << endl;      // c = 1.2

  return 0;
}
```

Шаг 4

```
#include <iostream>

class Double
{
  public:
  double number;
};

int main()
{
  using namespace std;

  Double a;
  a.number = 1.2;
  cout << "a = " << a.number << endl;  // a = 1.2
  Double b;
  b = a;
  cout << "b = " << b.number << endl;  // b = 1.2

  return 0;
}
```

Шаг 4. Версия 1

```
#include <iostream>

using namespace std;

class Double
{
  public:
  double number;

  Double& operator=(const Double&);
};

Double& Double::operator=(const Double& object)
{
  cout << "It's my copy assignment operator" << endl;
  if (this != &object) number = object.number;
  return *this;
}

int main()
{
  Double a;
  a.number = 1.2;
  cout << "a = " << a.number << endl;   // a = 1.2
  Double b;
  b = a;
  cout << "b = " << b.number << endl;   // b = 1.2

  return 0;
}
```