```cpp
#include <iostream>
#include <string>

class Software {
public:
    Software(int license, bool isSystem, std::string manuf, std::string ver)
        : license_no(license), is_system_software(isSystem), manufacturer(manuf),
version(ver) {}

    void print_info() const {
        std::cout << "License No: " << license_no << ", Manufacturer: " <<
manufacturer
                  << ", Version: " << version << std::endl;
    }

private:
    int license_no;
    bool is_system_software;
    std::string manufacturer;
    std::string version;
};

class Computer {
public:
    Computer(int serial, int ram, int rom, std::string processor)
        : serial_no(serial), RAM(ram), ROM(rom),
processor_manufacturer(processor) {}

    virtual void print_info() const {
        std::cout << "Serial No: " << serial_no << ", RAM: " << RAM
                  << ", ROM: " << ROM << ", Processor: " <<
processor_manufacturer << std::endl;
    }

private:
    int serial_no;
    int RAM;
    int ROM;
    std::string processor_manufacturer;
```

```cpp
};

class Laptop : public Computer {
public:
    Laptop(int serial, int ram, int rom, std::string processor, double
screenSize, double weight)
        : Computer(serial, ram, rom, processor), screen_size(screenSize),
weight(weight) {}

    void print_info() const override {
        Computer::print_info();
        std::cout << "Screen Size: " << screen_size << " inches, Weight: " <<
weight << " kg" << std::endl;
    }

private:
    double screen_size;
    double weight;
};

int main() {
    Computer* computers[10]; // Array of pointers to Computers

    // Initialize some computers and laptops
    computers[0] = new Computer(1, 8, 256, "Intel");
    computers[1] = new Laptop(2, 16, 512, "AMD", 15.6, 1.5);

    // Call print_info() for each computer
    for (int i = 0; i < 2; ++i) {
        computers[i]->print_info();
    }

    // Clean up memory
    for (int i = 0; i < 2; ++i) {
        delete computers[i];
    }
}
```

Resize code

```cpp
#include<iostream>
#include<string>
using namespace std;
void resize(int *p,int size, int newsize)
{
    if(size<newsize)
    {
        int *q= new int[newsize];
        for(int i=0;i<size;i++)
        {
            q[i]=p[i];}
            for(int i=size;i<newsize;i++)
            {q[i]=0;}

    }

}


int main()
{
    int size, newsize;
    cout<<"Enter size of array"<<endl;
    cin>>size;
    int *p=new int[size];
    cout<<"Enter the elements of array"<<endl;
    for (int i=0;i<size;i++)
    {cin>>p[i];
    }
    cout<<"Enter the new size of array"<<endl;
    cin>>newsize;
      resize(p,size,newsize);
    cout<<"Elements in array:";
     for (int i=0;i<newsize;i++)
            {
                cout<<p[i]<<" ";
            }

    cout<<endl;
    delete[] p;
```

```
    return 0;

}
```