```
#include<iostream>

using namespace std;


class String {
private:
  char *str;
  int length;
  int c_string_length(const char* s);
public:
  String();
  String(const char* s);
  String(const String& other);

  void print();

  String operator+(String other);
  bool operator==(String other);
  String& operator=(String& other);

  friend ostream& operator<<(ostream& os, String s);

  ~String();
};

int String::c_string_length(const char* s) {
  int i;
  for(i = 0; s[i] != '\0'; i++) {}
  return i;
}

String::String() {
  str = new char[1];
  str[0] = '\0';
  length = 0;
}

String::String(const char* s) {
  length = c_string_length(s);
  str = new char[length + 1];
  for(int i = 0; s[i] != '\0'; i++) {
    str[i] = s[i];
  }
  str[length] = '\0';
}

String::String(const String& other) {
  this->length = other.length;
  str = new char[this->length + 1];
  for(int i = 0; other.str[i] != '\0'; i++) {
    this->str[i] = other.str[i];
  }
  this->str[this->length] = '\0';
}

String String::operator+(String other) {
  char* s = new char[this->length + other.length + 1];
  for(int i = 0; i < this->length; i++) {
    s[i] = this->str[i];
  }
  for(int i = 0; i < other.length; i++) {
    s[i + this->length] = other.str[i];
  }
```

```
  s[this->length + other.length] = '\0';
  String ret = String(s);
  delete[] s;
  return ret;
}

void String::print() {
  cout << str << endl;
}

bool String::operator==(String other) {
  /*
  if(this->length != other.length) {
    return false;
  }
  */
  for(int i = 0; i < this->length + 1; i++) {
    if (this->str[i] != other.str[i]) {
      return false;
    }
  }
  return true;
}

String& String::operator=(String& other) {
  if(this == &other) {
    return *this;
  }
  delete[] this->str;
  this->str = new char[other.length + 1];
  for(int i = 0; i < other.length; i++) {
    this->str[i] = other.str[i];
  }
  this->str[other.length] = '\0';
  this->length = other.length;
  return *this;
}

String::~String() {
  delete[] str;
}

ostream& operator<<(ostream& os, String s) {
  cout << s.str;
  return os;
}



class Worker {
protected:
  String name;
  int pay;
public:
  Worker(String n, int p) : name(n), pay(p) {}
};


class Temp : public Worker {
protected:
  int end;
public:
  Temp(String n, int f, int l) : Worker(n, f), end(l) {}
  void extend(int n);
```

```
};

void Temp::extend(int n) {
  lejarat += n;
}


class Spec: public Worker {
protected:
  String field;
public:
  // TODO
}

int main(void) {
  String name = String("Bela");
  String name2 = String("Terez");
  Worker d1 = Worker(name, 20000);
  Temp i1 = Temp(name2, 30000, 2025);

  Worker* p = new Temp(name2, 30000, 2025);
  return 0;
}
```