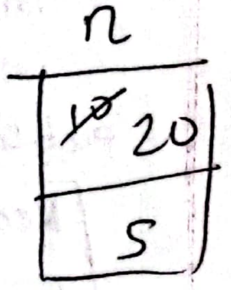
C++ Only

class and constructor

```
class Rectangle {
 private;
   int length;
   int breadth;
 public:          → constructor
  Rectangle (int l, int b) {
     length = l;
     breadth = b;
  }
  int area () {
     return length * breadth;
  }
  int changeLength (int l){
     Length = l;
  };
int main () {
  Rectangle r(10,5);
  r.area();
  r.changeLength(20);
}
```



initialize.
area()
changelgth.

class and constructor

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

class Rectangle {
   private:
      int length;
      int breadth;
   public:                                  → default constructor.
    [Rectangle(){length = breadth = 1;}.
    [Rectangle(int L, int b); → parameterized constructor
                                  constructor overloading
facilitator{int area();
           [int perimeter();
accessor [int getLength(){return length;}.
mutator. [void setLength (int l) {length=l;}.
      ~Rectangle(); → destructor
};

Rectangle:: Rectangle (int l, int){
   length = l;
   breadth=b;
}

int Rectangle::area(){
   return length * breadth;
}
```

```
int Rectangle:: Perimeter(){
    return 2* (length + breadth);
}
Rectangle :: ~Rectangle () {}

int main () {
  Rectangle r(10, 5);
  cout<< r.area();
  cout << r.perimeter();
  r.setLength (20);
  cout << r.getLength():
}
```