```
Script started on 2023-11-15 22:23:27+00:00 [TERM="xterm-256color" TTY="/dev/pts/0" COLUM
NS="59" LINES="54"]
\[\033[01;34m\]\w\[\033[00m\]$ pwd
/home/runner/Lab-18-Introduction-to-Overloading-Operators-kcp3s
\[\033[01;34m\]\w\[\033[00m\]$ cat -n main.cpp
     1  #include "Rectangle.h"
     2  #include <iostream>
     3
     4  int main() {
     5    Rectangle one{10, 12};
     6    Rectangle two{10, 23};
     7    Rectangle three{10, 12};
     8    // Less than
     9    if (one < three) {
    10      std::cout << "Rectangle one is less than three\n";
    11    } else {
    12      std::cout << "Rectangle one is greater than three\n";
    13    }
    14
    15    // Greater than
    16    if (two > three) {
    17      std::cout << "Rectangle two is greater than three\n";
    18    } else {
    19      std::cout << "Rectangle two is less than three\n";
    20    }
    21
    22    // Not equals
    23    if (one != two) {
    24      std::cout << "Rectangle one is not equal to two\n";
    25    } else {
    26      std::cout << "Rectangle one is equal to two\n";
    27    }
    28
    29    // Equals
    30    if (one == two) {
    31      std::cout << "Rectangle one is equal to two\n";
    32    } else {
    33      std::cout << "Rectangle one is not equal to two\n";
    34    }
    35
    36    // Lessthanequals
    37    if (one <= three) {
    38      std::cout << "Rectangle one is less than equals three\n";
    39    } else {
    40      std::cout << "Rectangle one is greater equals than three\n";
    41    }
    42
    43    // Greaterthanequals
    44    if (two > three) {
    45      std::cout << "Rectangle two is greater than equals three\n";
    46    } else {
    47      std::cout << "Rectangle two is less than equals three\n";
    48    }
    49
    50    // Prefix increment
    51    ++two;
    52    std::cout << "Two length " <<two.get_length() << " two width " << two.get_width
();
    53
    54    std::cout << std::endl;
    55
    56    // Postfix increment
    57    --three;
    58    std::cout << "Three length " << three.get_length() << " three width " << three.
get_width();
    59  }\[\033[01;34m\]\w\[\033[00m\]$ cat -n Rectangle.h
     1  #ifndef _RECTANGLE_H
     2  #define _RECTANGLE_H
     3
     4  class Rectangle {
```

```
     5  private:
     6    int length;
     7    int width;
     8  
     9  public:
    10    Rectangle(){};
    11    Rectangle(int l, int w);
    12    int get_length() const;
    13    int get_width() const;
    14    int get_area() const;
    15  
    16    bool operator<(Rectangle rhs);  // less than
    17    bool operator>(Rectangle rhs);  // greater than
    18    bool operator!=(Rectangle rhs); // not equal
    19    bool operator==(Rectangle rhs); // equal
    20    bool operator<=(Rectangle rhs); // less than or equal to
    21    bool operator>=(Rectangle rhs); // greater than or equal to
    22    void operator++();              // prefix increment
    23    void operator--();              // prefix decrement
    24  };
    25  
    26  #endif\[\033[01;34m\]\w\[\033[00m\]$ cat -n Rectangle.cpp
     1  // cpp code here
     2  #include "Rectangle.h"
     3  
     4  //Rectangle::Rectangle() {}
     5  Rectangle::Rectangle(int l, int w) {
     6    length = l;
     7    width = w;
     8  }
     9  int Rectangle::get_length() const { return length; }
    10  int Rectangle::get_width() const { return width; }
    11  int Rectangle::get_area() const { return (length * width); }
    12  
    13  bool Rectangle::operator<(Rectangle rhs) {
    14    bool lessthan = false;
    15    if (get_area() < rhs.get_area())
    16      lessthan = true;
    17    return lessthan;
    18  }
    19  bool Rectangle::operator>(Rectangle rhs) {
    20    bool greaterthan = false;
    21    if (get_area() > rhs.get_area())
    22      greaterthan = true;
    23    return greaterthan;
    24  }
    25  bool Rectangle::operator!=(Rectangle rhs) {
    26    bool notequals = false;
    27    if (get_area() != rhs.get_area())
    28      notequals = true;
    29    return notequals;
    30  }
    31  bool Rectangle::operator==(Rectangle rhs) {
    32    bool equals = false;
    33    if (get_area() == rhs.get_area())
    34      equals = true;
    35    return equals;
    36  }
    37  bool Rectangle::operator<=(Rectangle rhs) {
    38    bool lessthanequals = false;
    39    if (get_area() <= rhs.get_area())
    40      lessthanequals = true;
    41    return lessthanequals;
    42  }
    43  bool Rectangle::operator>=(Rectangle rhs) {
    44    bool greaterthanequals = false;
    45    if (get_area() >= rhs.get_area())
    46      greaterthanequals = true;
    47    return greaterthanequals;
    48  }
```

```
void Rectangle::operator++() {
  ++length;
  ++width;
}
void Rectangle::operator--() {
  --length;
  --width;
}\[\033[01;34m\]\w\[\033[00m\]$ g++ main.ppp Rectangle.cpp -o demo
\[\033[01;34m\]\w\[\033[00m\]$ ./demo
Rectangle one is greater than three
Rectangle two is greater than three
Rectangle one is not equal to two
Rectangle one is not equal to two
Rectangle one is less than equals three
Rectangle two is greater than equals three
Two length 11 two width 24
Three length 9 three width 11\[\033[01;34m\]\w\[\033[00m\]$ exit

Script done on 2023-11-15 22:24:29+00:00 [COMMAND_EXIT_CODE="0"]
```