CS 300 Data Structures
Problem Set 7 – Big Rule of Three

Consider the following IntCell class definition.

```
#ifndef INT_CELL_H
#define INT_CELL_H

class IntCell{
    private:
        int* x;
    public:
        IntCell(int);
        IntCell(const IntCell& other);              //copy constructor
        int read() const;
        void write(int);
        IntCell& operator=(const IntCell& other); //copy assignment operator
        ~IntCell();                               //destructor
        IntCell operator+(int);                  //operator+ overloading
};
#endif
```

```
#include "IntCell.h"
#include <iostream>
using namespace std;

IntCell::IntCell(int _x){
    cout<<"constructor"<<endl;
    x = new int(_x);
}

IntCell::IntCell(const IntCell& other){
    cout<<"copy constructor"<<endl;
    x = NULL;
    if(other.x != NULL){
        x = new int(*(other.x));
    }
}

int IntCell::read() const{
    return *x;
}

void IntCell::write(int _x){
    *x = _x;
}

IntCell IntCell::operator+(int a){
    cout<<"operator overloading"<<endl;
    *x += a;

    return *this;
}
IntCell& IntCell::operator=(const IntCell& other){
    cout<<"assignment"<<endl;
    delete x;
    x = new int(*(other.x));
    return *this;
}

IntCell::~IntCell(){
    cout<<"destructor"<<endl;
    delete x;
}
```

1. What is the output of the following program?

```
#include <iostream>
#include "IntCell.h"

using namespace std;

int main(){
    IntCell cell(10);

    cout<<cell.read()<<endl;

    return 0;
}
```

```
constructor
10
destructor
```

2. What is the output of the following program?

```
#include <iostream>
#include "IntCell.h"
```

```
using namespace std;

void print(IntCell& cell);

int main(){
    IntCell cell(10);

    print(cell);

    return 0;
}

void print(IntCell& cell){
    cout<<cell.read()<<endl;
}
```

```
constructor
10
destructor
```

3. What is the output of the following program?

```
#include <iostream>
#include "IntCell.h"

using namespace std;

void print(IntCell cell);

int main(){
    IntCell cell(10);

    print(cell);

    return 0;
}

void print(IntCell cell){
    cout<<cell.read()<<endl;
}
```

```
constructor
copy constructor
```

```
10
destructor
destructor
```

4.

```
    #include <iostream>
    #include "IntCell.h"

    using namespace std;

    IntCell print(IntCell cell);

    int main(){
        IntCell cell(10);

        IntCell cell2 =print(cell);

        return 0;
    }

    IntCell print(IntCell cell){
        cout<<cell.read()<<endl;
        return IntCell(cell);
    }
```

```
constructor
copy constructor
10
copy constructor
destructor
destructor
destructor
```

4. What is the output of the following program?

```
#include <iostream>
#include "IntCell.h"

using namespace std;

void print(IntCell);

int main(){
    IntCell cell(10);
    print(cell);
    cell = cell + 3; //(cell.operator+(int))

    return 0;
}

void print(IntCell cell){
    cout<<cell.read()<<endl;
}
```

Answer:
constructor
copy constructor
10
destructor
operator overloading
copy constructor
assignment
destructor
destructor