General Tips on Programming

- Write your code modularly
- top-down approach
- Compile + test functionality "on the fly"
- Start with an "empty" program/classes
- Add content gradually and keep testing
- If something goes wrong, probably the bug is in the latest change...

Even More Tips

- Based on a true story:
  - Do not send us code
  - Do not send us emails saying "the code you gave us does not work", before you make sure the original given code is problematic
  - When you ask questions be as specific as you can, give as much information about the problem and your trials as possible
Customer

```java
public class Customer {
    public Customer(String name, String id) {
        this.name = name;
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public String getID() {
        return id;
    }
    private String name;
    private String id;
}
```

Toy Bank Program

```java
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);
        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);
        System.out.println("account 1 has " + account1.getBalance());
        System.out.println("account 2 has " + account2.getBalance());
    }
}
```

Object Diagram

```
Customer: String name: String id: String: "Avi Cohen"
Customer: String name: String id: String: "Rita Stein"
Bank
main (…) {
    customer1: customer2:
}
```
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);
        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(account2, account3);
        account2.withdraw(300);
        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}

public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);
        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(account2, account3);
        account2.withdraw(300);
        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}

public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);
        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(account2, account3);
        account2.withdraw(300);
        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}

Output

public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);
        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(account2, account3);
        account2.withdraw(300);
        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
public class MP3Song {
    public void play() {
        // audio codec calculations, play the song...
    }
    // does complicated stuff related to MP3 format....
}

public class Player {
    private boolean repeat;
    private boolean shuffle;
    public void playSongs(MP3Song[] songs) {
        do {
            if (shuffle)
                Collections.shuffle(Arrays.asList(songs));
            for (MP3Song song : songs)
                song.play();
        } while (repeat);
    }
}

public class VideoClip {
    public void play() {
        // video codec calculations, play the clip...
    }
    // does complicated stuff related to video formats....
}

public class VideoClip implements playable {
    @Override
    public void play() {
        // render video, play the clip on screen...
    }
    // does complicated stuff related to video formats....
}

public class MP3Song implements playable {
    @Override
    public void play() {
        // audio codec calculations, play the song...
    }
    // does complicated stuff related to MP3 format....
}

public class MP4Song implements playable {
    @Override
    public void play() {
        // audio codec calculations, play the song...
    }
    // does complicated stuff related to MP3 format....
}
פונקציות על סיביות

פעולה על סיביות

- Unary bitwise complement
- Signed left shift
- Signed right shift
- Unsigned right shift
- Bitwise AND
- Bitwise XOR
- Bitwise OR

32 bit int

מה נקבל מ 3 & 3? ómo נקבל מ0x30 & 1?