Asynchronize Programming

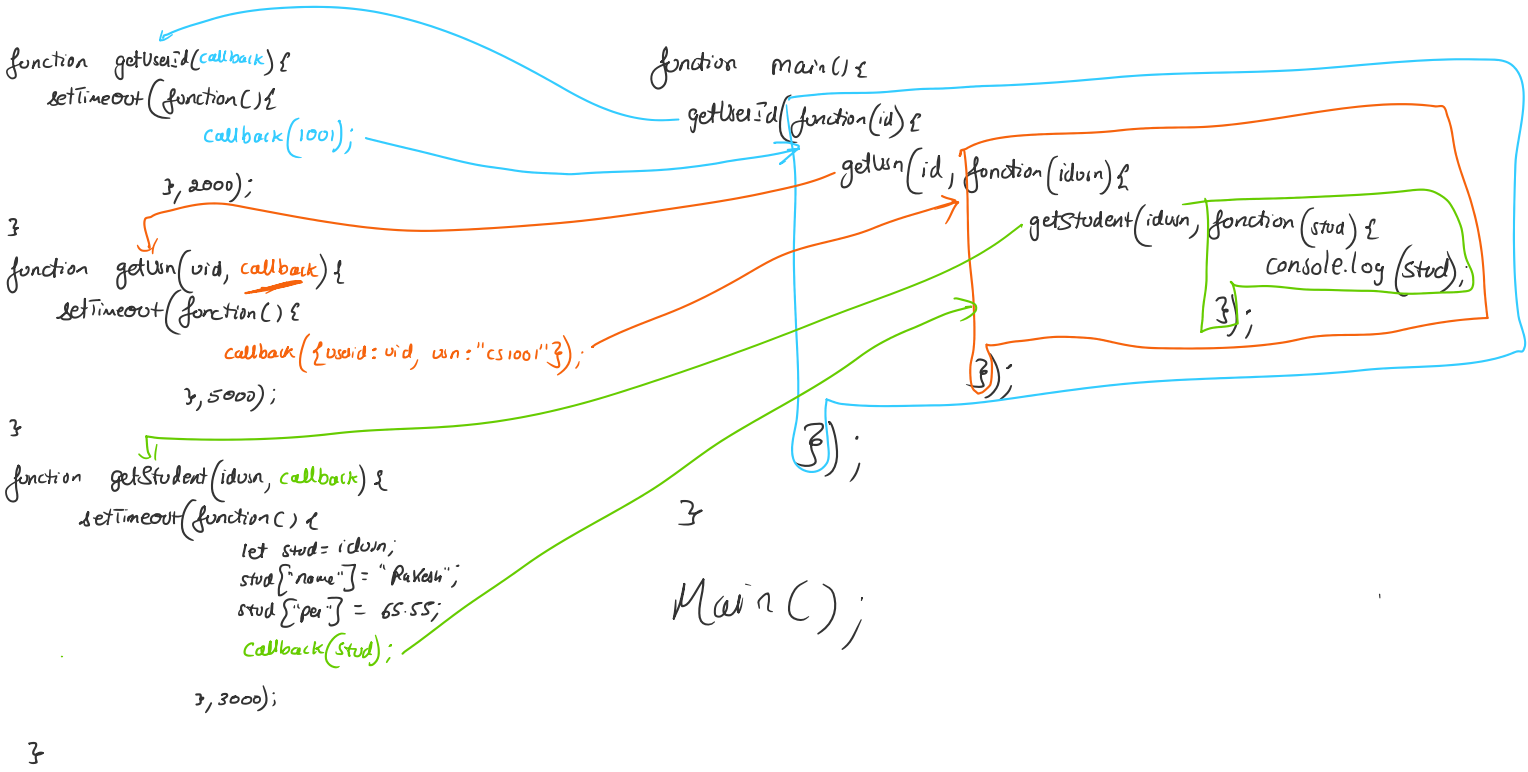
```
setTimeOut( function( ){
      ≡
})
```

```
setTimeout( function(){

}, millisec)
```

```
function display(msg){
    s.o.p(" Start of display method");
    setTimeout (function (){
                    console.log(msg);
          }, 5000);
    s.o.p (" End of display method");
}
display("Hello world");
console.log (" End of Main");
```

Start of display method
End of display method
End of Main
Hello world


function getUserId(callback){
setTimeout(function(){
callback(1001);
},2000);
}
function getUsn(uid, callback){
setTimeout(function(){
callback({usrid:uid, usn:"CS1001"});
},5000);
}
function getStudent(idusn, callback){
setTimeout(function(){
let stud=idusn;
stud["name"]="Rakesh";
stud["per"]=65.55;
callback(stud);
},3000);
}
function main(){
getUserId(function(id){
getUsn(id, function(idusn){
getStudent(idusn, function(stud){
console.log(stud);
});
});
});
}
main();

Promise

```
function getMessage(callback) {
    setTimeout(() => {
                callback("Hello world");
            }, 5000);
}
function Main() {
    getMessage((msg) => {
                console.log(msg);
            });
}
Main();
```

Promise(resolve, reject)

```
function getMessage() {
    let p = new Promise((resolve, reject) => {
                setTimeout(() => {
                        resolve("Hello world");
                    }, 5000);
            });
    return(p);
}
function Main() {
    let p = getMessage();
    p.then((msg) => console.log(msg));
}
Main();
```

```
function evenodd(N) {
    let promise = new Promise((resolve, reject) => {
                setTimeout(() => {
                        if(N%2==0)
                            resolve();
                        else
                            reject();
                    });
            });

    return promise;
}

function Main() {
    let p = evenodd(8);
    p.then(() => console.log("No is even"))
     .catch(() => console.log("No is odd"));
}
```