**Name : Mariyam Mahnoor**

**Course :Blockchain Programming**

**Section:“B”**

**Date: 8-Dec-2020**

## Project Management Tool:

```
//
define
assignee
class
           class Assignee{
               constructor(name ,title,status){
                   this.name= name
                   this.title= title
                   this.status= status
               }
               ChangeStatus(value){
                   this.status= value
               }
           }
           // creating objects with assignee class
           var hamza= new Assignee('Hamza khan','finance Manager','active')
           var musab= new Assignee('Musab khan','machanical Engineer','active')
           var Ali= new Assignee('ali khan','marketing Manager','active')
           var Mustafain = new Assignee('Mustafain khan','web designer','fresher')
           var owais= new Assignee('Owais khan',' graphic Designer','active')
           // define tasks class
           class tasks{
                      constructor(name,days,AssigneeId,taskstatus){
                                 this.tname = name
                   this.tdays = days
                   this.AssigneeId= AssigneeId
                   this.taskstatus= taskstatus
               }
               ChangeTaskStatus(value){
                   this.taskstatus= value
               }
           }
           // creating objects with tasks class
           var task1 = new tasks('architecture',5,hamza,'inprogress')
           var task2 = new tasks('development_1',10,Ali,'notstarted')
           var task3 = new tasks('development_2',4,musab,'done')
           var task4 = new tasks('development_3',15,owais,'inprogress')
           // creating map for multiple tasks
           var blockchain_task = new Map();
           blockchain_task.set('1',task1);
           blockchain_task.set('2',task2);
           blockchain_task.set('3',task3);
           blockchain_task.set('4',task4);
           // define project  class
           class project{
                      constructor(name,type,days,tasks,dayscompleted){
                   this.tasks = tasks
                   this.status="in progress";
                   this.days = days
```

```
    this.name=name
    this.type = type
    this.dayscompleted = dayscompleted
       }
       print(){
                  return this.tasks
       }
       totaltasks(){
                  return this.tasks.size
}
daysRemaining(){
    var remainingday=this.dayscompleted-this.days
    return remainingday

}
status_(){
    var i=0;
    var j=0;
    var k=0;
    for (let [key, value] of this.tasks.entries()) {


        if(value.taskstatus =="done"){

            console.log(key + ' = ' + value.taskstatus)
            i+=1;

        }

        else if(value.taskstatus =="inprogress"){

            console.log(key + ' = ' + value.taskstatus)
            j+=1;

        }
        else{

            console.log(key + ' = ' + value.taskstatus)
            k+=1;

        }


      }
      console.log(`Completed task ${i}`)
      console.log(`Inprogress task ${j}`)
      console.log(`Not started task ${k}`)
      if(i==this.tasks.size){
          console.log("project completed")
      }
      else if(i < this.tasks.size){
        console.log("Project  is inprogress")
    }
    else if(k == this.tasks.size){
        console.log("project not started")
    }
    else{
        console.log("not define")
    }

}
CompletedTasks(){
    console.log("Completed task")
    for (let [key, value] of this.tasks.entries()) {
            if(value.taskstatus =="done"){

                console.log(key + ' = ' + value.taskstatus)
            }

          }
          console.log('\n---------------')
    }
    PendingTasks(){
        console.log("Pending task")
        for (let [key, value] of this.tasks.entries()) {

            if(value.taskstatus!="done"){

                console.log(key + ' = ' + value.taskstatus)
            }

          }
          console.log('\n---------------')

    }
    TaskAssigneee(taskid) {
    }
           changeTask(taskid,task){
        this.tasks.set(taskid,task)
                      console.log(this.tasks.get(taskid,task))
    }
    DeleteTask(value){
        this.tasks.delete(value)
    }
    AddTask(key,value){
    this.tasks.set(key,value)
    }
    ChangeAssignee(taskid,value){
        this.tasks.get(taskid).AssigneeId= value;
    }
}
var realstate_blockchain = new project('blockchain','software',15,blockchain_task,'50')
console.log(realstate_blockchain.print())
console.log(realstate_blockchain.totaltasks())
console.log("change task")
realstate_blockchain.changeTask('1',task2)
realstate_blockchain.changeTask('2',task1)
console.log(realstate_blockchain.print())
realstate_blockchain.CompletedTasks()
realstate_blockchain.PendingTasks()
console.log(realstate_blockchain.daysRemaining())
realstate_blockchain.status_()
realstate_blockchain.ChangeAssignee('1',Mustafain)
console.log(realstate_blockchain.print())
```

## Output:

```
Map {

 '1' => tasks {

  tname: 'architecture',

  tdays: 5,

  AssigneeId: Assignee {

   name: 'Hamza khan',
    title: 'finance Manager',
    status: 'active'
  },
  taskstatus: 'inprogress'
},
'2' => tasks {
  tname: 'development_1',
  tdays: 10,
  AssigneeId: Assignee {
    name: 'ali khan',
    title: 'marketing Manager',
    status: 'active'
  },
  taskstatus: 'notstarted'
},
'3' => tasks {
  tname: 'development_2',
  tdays: 4,
  AssigneeId: Assignee {
    name: 'Musab khan',
    title: 'machanical Engineer',
    status: 'active'
  },
  taskstatus: 'done'
},
'4' => tasks {
  tname: 'development_3',
  tdays: 15,
  AssigneeId: Assignee {
    name: 'Owais khan',
    title: ' graphic Designer',
      status: 'active'
    },
    taskstatus: 'inprogress'
  }
}
4
change task
tasks {
  tname: 'development_1',
  tdays: 10,
  AssigneeId: Assignee {
    name: 'ali khan',
    title: 'marketing Manager',
    status: 'active'
  },
  taskstatus: 'notstarted'
}
tasks {
  tname: 'architecture',
  tdays: 5,
  AssigneeId: Assignee {
    name: 'Hamza khan',
    title: 'finance Manager',
    status: 'active'
  },
  taskstatus: 'inprogress'
}
Map {
  '1' => tasks {
    tname: 'development_1',
    tdays: 10,
  AssigneeId: Assignee {
    name: 'ali khan',
    title: 'marketing Manager',
    status: 'active'
  },
  taskstatus: 'notstarted'
},
'2' => tasks {
  tname: 'architecture',
  tdays: 5,
  AssigneeId: Assignee {
    name: 'Hamza khan',
    title: 'finance Manager',
    status: 'active'
  },
  taskstatus: 'inprogress'
},
'3' => tasks {
  tname: 'development_2',
  tdays: 4,
  AssigneeId: Assignee {
    name: 'Musab khan',
    title: 'machanical Engineer',
    status: 'active'
  },
  taskstatus: 'done'
},
'4' => tasks {
  tname: 'development_3',
  tdays: 15,
  AssigneeId: Assignee {
   name: 'Owais khan',
   title: ' graphic Designer',
   status: 'active'
  },
  taskstatus: 'inprogress'
 }
}
Completed task
3 = done

---------------
Pending task
1 = notstarted
2 = inprogress
4 = inprogress

---------------
35
1 = notstarted
2 = inprogress
3 = done
4 = inprogress
Completed task 1
Inprogress task 2
Not started task 1
Project  is inprogress
Map {
 '1' => tasks {
  tname: 'development_1',
  tdays: 10,
  AssigneeId: Assignee {
    name: 'Mustafain khan',
    title: 'web designer',
    status: 'fresher'
  },
  taskstatus: 'notstarted'
},
'2' => tasks {
  tname: 'architecture',
  tdays: 5,
  AssigneeId: Assignee {
    name: 'Hamza khan',
    title: 'finance Manager',
    status: 'active'
  },
  taskstatus: 'inprogress'
},
'3' => tasks {
  tname: 'development_2',
  tdays: 4,
  AssigneeId: Assignee {
    name: 'Musab khan',
    title: 'machanical Engineer',
    status: 'active'
  },
  taskstatus: 'done'
},
'4' => tasks {
  tname: 'development_3',
  tdays: 15,
  AssigneeId: Assignee {
    name: 'Owais khan',
   title: ' graphic Designer',
   status: 'active'
  },
  taskstatus: 'inprogress'
 }
}
```