

DAY 4 (03-08-2023)

1. Learnt about MongoDB by using MongoSh :

Microsoft Windows [Version 10.0.22621.1992]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Sriram>mongosh

Current Mongosh Log ID: 64cb30805c9aae408784dbd6

Connecting to:

mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.3

Using MongoDB: 6.0.8

Using Mongosh: 1.10.3

For mongosh info see: <https://docs.mongodb.com/mongodb-shell/>

The server generated these startup warnings when booting

2023-08-02T13:05:57.323+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

test> show dbs

admin 40.00 KiB

config 60.00 KiB

local 40.00 KiB

test> show dbs

admin 40.00 KiB

config 72.00 KiB

local 40.00 KiB

sample 8.00 KiB

test> db

test

test> db payoda

Uncaught:

SyntaxError: Missing semicolon. (1:2)

```
> 1 | db payoda
```

```
    | ^
```

```
2 |
```

```
test> db payoda;
```

Uncaught:

SyntaxError: Missing semicolon. (1:2)

```
> 1 | db payoda;
```

```
    | ^
```

```
2 |
```

```
test> db;
```

```
test
```

```
test> use payoda
```

```
switched to db payoda
```

```
payoda> db.createCollection("employees")
```

```
{ ok: 1 }
```

```
payoda> db.employees.insertOne({name:"Sriram"})
```

```
{
```

```
  acknowledged: true,
```

```
  insertedId: ObjectId("64cb3d275c9aae408784dbd7")
```

```
}
```

```
payoda> db.employees.insertMany({name:"Varun"},{name:"Yogesh"})
```

MongoInvalidArgumentError: Argument "docs" must be an array of documents

```
payoda> db.employees.insertMany({{name:"Varun"},{name:"Yogesh"}})
```

Uncaught:

SyntaxError: Unexpected token (1:25)

```
> 1 | db.employees.insertMany({{name:"Varun"},{name:"Yogesh"}})
      |           ^
2 |
```

```
payoda> db.employees.insertMany([{name:"Varun"},{name:"Yogesh"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64cb3df65c9aae408784dbd8"),
    '1': ObjectId("64cb3df65c9aae408784dbd9")
  }
}
```

```
payoda> db.employees.find()
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' }
]
```

```
payoda> db.employees.find({name:"Sriram"})
[ { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' } ]
payoda> db.employees.find({name:"sriram"})
```

```
payoda> db.employees.find({Name:"Sriram"})
```

```
payoda> db.employees.insertOne({name:"Sriram"})
{
  acknowledged: true,
  insertedId: ObjectId("64cb3f895c9aae408784dbda")
}
payoda> db.employees.find()
```

```

[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' }
]

payoda> db.employees.insertMany([{name:"Nitish"},{name:"Preethi", age: "30"}])

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64cb3fce5c9aae408784dbdb"),
    '1': ObjectId("64cb3fce5c9aae408784dbdc")
  }
}

payoda> db.employees.find()

[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  }
]

payoda> db.employees.find({}, {name:1})

[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },

```

```
{_id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
{ _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
{ _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
{ _id: ObjectId("64cb3fce5c9aae408784dbdc"), name: 'Preethi' }
]
```

```
payoda> db.employees.find({}, {name:0})
```

```
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7") },
  { _id: ObjectId("64cb3df65c9aae408784dbd8") },
  { _id: ObjectId("64cb3df65c9aae408784dbd9") },
  { _id: ObjectId("64cb3f895c9aae408784dbda") },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb") },
  { _id: ObjectId("64cb3fce5c9aae408784dbdc"), age: '30' }
]
```

```
payoda> db.employees.insertMany([ {name:"Sanjay"}, {name:"Harsha", age: 30} ])
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64cb40105c9aae408784dbdd"),
    '1': ObjectId("64cb40105c9aae408784dbde")
  }
}
```

```
payoda> db.employees.find()
```

```
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
  }
]
```

```
    name: 'Preethi',
    age: '30'
  },
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
    _id: ObjectId("64cb40105c9aae408784dbde"),
    name: 'Harsha',
    age: 30
  }
]
```

```
payoda> db.employees.insertOne({name:JK})
```

ReferenceError: JK is not defined

```
payoda> db.employees.insertOne({name:"JK"})
```

```
{
  acknowledged: true,
  insertedId: ObjectId("64cb40505c9aae408784dbdf")
}
```

```
payoda> db.employees.find()
```

```
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  },
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
```

```

    _id: ObjectId("64cb40105c9aae408784dbde"),
    name: 'Harsha',
    age: 30
  },
  { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' }
]
payoda> db.employees.insertOne({name:J})
ReferenceError: J is not defined
payoda> db.employees.updateOne({name:"Sriram"},{$set:{name:"Sri"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
payoda> db.employees.find()
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sri' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  },
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
    _id: ObjectId("64cb40105c9aae408784dbde"),

```

```

    name: 'Harsha',
    age: 30
  },
  { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' }
]
payoda> db.employees.updateOne({},{$set:{name:"Sri"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
payoda> db.employees.find()
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sri' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  },
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
    _id: ObjectId("64cb40105c9aae408784dbde"),
    name: 'Harsha',
    age: 30
  },

```



```

    { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' }
  ]
payoda> db.employees.updateOne({name:"Sriram"},{$set:{name:"Sri"}},{upsert:true})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
payoda> db.employees.find()
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sri' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sri' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  },
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
    _id: ObjectId("64cb40105c9aae408784dbde"),
    name: 'Harsha',
    age: 30
  },
  { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' }
]
payoda> db.employees.updateOne({name:"Sriram"},{$set:{name:"Sri"}},{upsert:true})

```

```

{
  acknowledged: true,
  insertedId: ObjectId("64cb4984b683a6a089140de4"),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
payoda> db.employees.find()
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sri' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sri' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  },
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
    _id: ObjectId("64cb40105c9aae408784dbde"),
    name: 'Harsha',
    age: 30
  },
  { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' },
  { _id: ObjectId("64cb4984b683a6a089140de4"), name: 'Sri' }
]
payoda> db.employees.updateMany({name:"Sri"},{$set:{name:"Sriram"}},{upsert:true})
{
  acknowledged: true,

```

```

    insertedId: null,
    matchedCount: 3,
    modifiedCount: 3,
    upsertedCount: 0
  }
}

payoda> db.employees.find()

[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  },
  {
    _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
    _id: ObjectId("64cb40105c9aae408784dbde"),
    name: 'Harsha',
    age: 30
  },
  { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' },
  { _id: ObjectId("64cb4984b683a6a089140de4"), name: 'Sriram' }
]

payoda> db.employees.updateMany({name:"AAA"},{$set:{name:"Sriram"}},{upsert:true})

{
  acknowledged: true,
  insertedId: ObjectId("64cb4a0db683a6a089140e41"),
  matchedCount: 0,

```

```

    modifiedCount: 0,
    upsertedCount: 1
  }
payoda> db.employees.updateMany({name:"AAA"},{$set:{name:"Sriram"}},{upsert:true})
{
  acknowledged: true,
  insertedId: ObjectId("64cb4a0fb683a6a089140e43"),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
payoda> db.employees.find()
[
  { _id: ObjectId("64cb3d275c9aae408784dbd7"), name: 'Sriram' },
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
  {
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),
    name: 'Preethi',
    age: '30'
  },
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
  {
    _id: ObjectId("64cb40105c9aae408784dbde"),
    name: 'Harsha',
    age: 30
  },
  { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' },
  { _id: ObjectId("64cb4984b683a6a089140de4"), name: 'Sriram' },

```

```
{ _id: ObjectId("64cb4a0db683a6a089140e41"), name: 'Sriram' },  
{ _id: ObjectId("64cb4a0fb683a6a089140e43"), name: 'Sriram' }  
]
```

```
payoda> db.employees.deleteOne({name:"Sriram"})
```

```
{ acknowledged: true, deletedCount: 1 }
```

```
payoda> db.employees.find()
```

```
[  
  { _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },  
  { _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },  
  { _id: ObjectId("64cb3f895c9aae408784dbda"), name: 'Sriram' },  
  { _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },  
  {  
    _id: ObjectId("64cb3fce5c9aae408784dbdc"),  
    name: 'Preethi',  
    age: '30'  
  },  
  { _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },  
  {  
    _id: ObjectId("64cb40105c9aae408784dbde"),  
    name: 'Harsha',  
    age: 30  
  },  
  { _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' },  
  { _id: ObjectId("64cb4984b683a6a089140de4"), name: 'Sriram' },  
  { _id: ObjectId("64cb4a0db683a6a089140e41"), name: 'Sriram' },  
  { _id: ObjectId("64cb4a0fb683a6a089140e43"), name: 'Sriram' }  
]
```

```
payoda> db.employees.deleteMany({name:"Sriram"})
```

```
{ acknowledged: true, deletedCount: 4 }
```

```
payoda> db.employees.find()
```

```
[
```

```

{ _id: ObjectId("64cb3df65c9aae408784dbd8"), name: 'Varun' },
{ _id: ObjectId("64cb3df65c9aae408784dbd9"), name: 'Yogesh' },
{ _id: ObjectId("64cb3fce5c9aae408784dbdb"), name: 'Nitish' },
{
  _id: ObjectId("64cb3fce5c9aae408784dbdc"),
  name: 'Preethi',
  age: '30'
},
{ _id: ObjectId("64cb40105c9aae408784dbdd"), name: 'Sanjay' },
{
  _id: ObjectId("64cb40105c9aae408784dbde"),
  name: 'Harsha',
  age: 30
},
{ _id: ObjectId("64cb40505c9aae408784dbdf"), name: 'JK' }
]

```

```
payoda> db.employees.deleteMany()
```

```
MongoshInvalidInputError: [COMMON-10001] Missing required argument at position 0
(Collection.deleteMany)
```

```
payoda> db.employees.deleteMany({})
```

```
{ acknowledged: true, deletedCount: 7 }
```

```
payoda> db.employees.find()
```

```
payoda> db.employees.insertOne({name: "Sriram", age: 22})
```

```

{
  acknowledged: true,
  insertedId: ObjectId("64cb4b2d5c9aae408784dbe0")
}

```

```
payoda> db.employees.insertOne({name: "Varun", age: 22})
```

```

{
  acknowledged: true,

```

```

    insertedId: ObjectId("64cb4b4e5c9aae408784dbe1")
  }
payoda> db.employees.updateOne({name:"Sriram"},{$inc: {age:10}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
payoda> db.employees.find()
[
  {
    _id: ObjectId("64cb4b2d5c9aae408784dbe0"),
    name: 'Sriram',
    age: 32
  },
  { _id: ObjectId("64cb4b4e5c9aae408784dbe1"), name: 'Varun', age: 22 }
]
payoda> db.employees.updateOne({name:"Sriram"},{$dec: {age:10}})
MongoServerError: Unknown modifier: $dec. Expected a valid update modifier or pipeline-style
update specified as an array
payoda> db.employees.updateOne({name:"Sriram"},{$set: {age:10}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
payoda> db.employees.find()

```

```
[
  {
    _id: ObjectId("64cb4b2d5c9aae408784dbe0"),
    name: 'Sriram',
    age: 10
  },
  { _id: ObjectId("64cb4b4e5c9aae408784dbe1"), name: 'Varun', age: 22 }
]
```

```
payoda> db.users.insertOne({
...   name: "John Doe",
...   age: 30,
...   email: "john.doe@example.com",
...   isActive: true
... });
{
  acknowledged: true,
  insertedId: ObjectId("64cb4cb65c9aae408784dbe2")
}
```

```
payoda> db.users.find()
[
  {
    _id: ObjectId("64cb4cb65c9aae408784dbe2"),
    name: 'John Doe',
    age: 30,
    email: 'john.doe@example.com',
    isActive: true
  }
]
```

```
payoda> db.users.find({age: {$gt: 25}})
[
  {
```



```
  _id: ObjectId("64cb4cb65c9aae408784dbe2"),
  name: 'John Doe',
  age: 30,
  email: 'john.doe@example.com',
  isActive: true
}
]
```

```
payoda> db.users.find({age: {$gt: 35}})
```

```
payoda> db.users.find({age: {$gt: 25}}, {name: 1});
```

```
[ { _id: ObjectId("64cb4cb65c9aae408784dbe2"), name: 'John Doe' } ]
```

```
payoda> db.users.find({age: {$gt: 25}}, {name: 1, age: 0});
```

MongoServerError: Cannot do exclusion on field age in inclusion projection

```
payoda> db.users.find({age: {$gt: 25}}, {age: 0});
```

```
[
  {
    _id: ObjectId("64cb4cb65c9aae408784dbe2"),
    name: 'John Doe',
    email: 'john.doe@example.com',
    isActive: true
  }
]
```

```
payoda> db.users.find({age: {$gt: 25}}, {id: 0});
```

```
[
  {
    _id: ObjectId("64cb4cb65c9aae408784dbe2"),
    name: 'John Doe',
    age: 30,
    email: 'john.doe@example.com',
    isActive: true
  }
]
```

```

]
payoda> db.users.find({age: {$gt: 25}},{_id: 0});
[
  {
    name: 'John Doe',
    age: 30,
    email: 'john.doe@example.com',
    isActive: true
  }
]
payoda> db.users.find({age: {$gt: 25}},{_id: 0,name: 1});
[ { name: 'John Doe' } ]
payoda> db.users.find({age: {$gt: 25}},{age: 0,name: 1});
MongoServerError: Cannot do inclusion on field name in exclusion projection
payoda> db.users.insertOne({ name: "John Doe", age: 30, email: "john.doe@example.com",
isActive: false });
{
  acknowledged: true,
  insertedId: ObjectId("64cb4f3a5c9aae408784dbe3")
}
payoda> db.users.insertOne({ name: "John Wick", age: 30, email: "john.doe@example.com",
isActive: false });
{
  acknowledged: true,
  insertedId: ObjectId("64cb4f495c9aae408784dbe4")
}
payoda> db.users.insertOne({ name: "John Wick", age: 30, email: "john.jk@example.com",
isActive: false });
{
  acknowledged: true,
  insertedId: ObjectId("64cb4f555c9aae408784dbe5")
}

```

```
payoda> db.users.find()

[
  {
    _id: ObjectId("64cb4cb65c9aae408784dbe2"),
    name: 'John Doe',
    age: 30,
    email: 'john.doe@example.com',
    isActive: true
  },
  {
    _id: ObjectId("64cb4f3a5c9aae408784dbe3"),
    name: 'John Doe',
    age: 30,
    email: 'john.doe@example.com',
    isActive: false
  },
  {
    _id: ObjectId("64cb4f495c9aae408784dbe4"),
    name: 'John Wick',
    age: 30,
    email: 'john.doe@example.com',
    isActive: false
  },
  {
    _id: ObjectId("64cb4f555c9aae408784dbe5"),
    name: 'John Wick',
    age: 30,
    email: 'john.jk@example.com',
    isActive: false
  }
]
```

```
payoda> db.users.find({name: "John Doe",{age: {$gt : 25}}})
```

Uncaught:

SyntaxError: Unexpected token (1:32)

```
> 1 | db.users.find({name: "John Doe",{age: {$gt : 25}}})
```

```
    |                ^
```

```
2 |
```

```
payoda> db.users.find({name: "John Doe",age: {$gt : 25}})
```

```
[
  {
    _id: ObjectId("64cb4cb65c9aae408784dbe2"),
    name: 'John Doe',
    age: 30,
    email: 'john.doe@example.com',
    isActive: true
  },
  {
    _id: ObjectId("64cb4f3a5c9aae408784dbe3"),
    name: 'John Doe',
    age: 30,
    email: 'john.doe@example.com',
    isActive: false
  }
]
```

```
payoda> db.users.find({name: "John Doe",age: {$gt : 25},isActive: true})
```

```
[
  {
    _id: ObjectId("64cb4cb65c9aae408784dbe2"),
    name: 'John Doe',
    age: 30,
```

```

    email: 'john.doe@example.com',
    isActive: true
  }
]

payoda> db.users.find({name: "John Doe",age: {$gt : 25},isActive: true},{name: 1})
[ { _id: ObjectId("64cb4cb65c9aae408784dbe2"), name: 'John Doe' } ]

payoda> db.users.find({name: "John Doe",age: {$lt : 35},isActive: true},{name: 1})
[ { _id: ObjectId("64cb4cb65c9aae408784dbe2"), name: 'John Doe' } ]

payoda> db.users.find({name: "John Doe",age: {$in : [25,30]},isActive: true},{name: 1})
[ { _id: ObjectId("64cb4cb65c9aae408784dbe2"), name: 'John Doe' } ]

payoda> db.users.find({name: "John Doe",age: {$in : [25,33]},isActive: true},{name: 1})

payoda> db.users.find({name: "John Doe",age: {$lt :33,$gt :20},isActive: true},{name: 1})
[ { _id: ObjectId("64cb4cb65c9aae408784dbe2"), name: 'John Doe' } ]

payoda> db.users.find({name: "John Doe",age: {$lt :33,$gt :20},isActive: true},{_id:0,name: 1})
[ { name: 'John Doe' } ]

payoda> db.users.find({$or: [{name: "John Doe",age: {$lt :33,$gt :20},isActive:
true}},{_id:0,name: 1})
[ { name: 'John Doe' } ]

payoda> db.users.find({$or: [{name: "John Wick",age: {$lt :33,$gt :20},isActive:
true}},{_id:0,name: 1})

Uncaught:
SyntaxError: Unexpected token, expected "," (1:82)

> 1 | db.users.find({$or: [{name: "John Wick"},{age: {$lt :33,$gt :20}},{isActive:
true}]}},{_id:0,name: 1})
    |                                     ^
2 |

```

```
payoda> db.users.find({$or: [{name: "John Wick"},{age: {$lt :33,$gt :20}},{isActive: true}]],{_id:0,name: 1})
```

```
[
  { name: 'John Doe' },
  { name: 'John Doe' },
  { name: 'John Wick' },
  { name: 'John Wick' }
]
```

```
payoda> db.users.find({$or: [{name: "John Wick"},{age: {$lte :30,$gt :20}},{isActive: true}]],{_id:0,name: 1})
```

```
[
  { name: 'John Doe' },
  { name: 'John Doe' },
  { name: 'John Wick' },
  { name: 'John Wick' }
]
```

```
payoda> payoda.users.find({$or: [{name: "John Wick"},{age: {$lte :30,$gt :20}},{isActive: true}]],{_id:0,name: 1})
```

ReferenceError: payoda is not defined

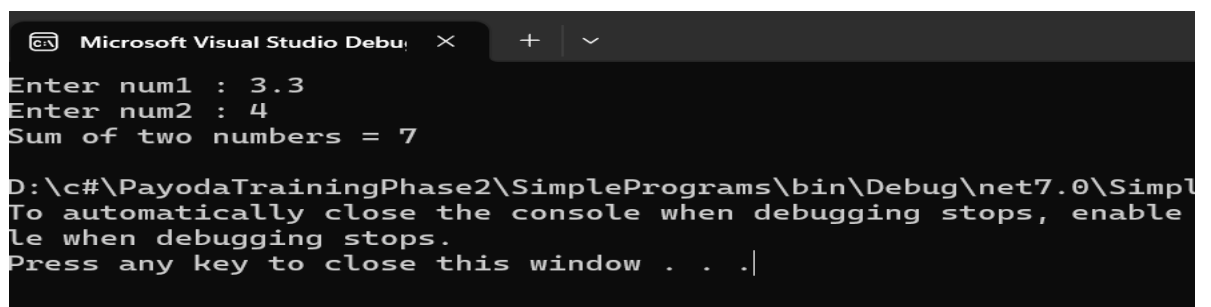
```
payoda>
```

2. C# fundamentals :

Addition:

```
double num1, num2;
int result;
Console.Write("Enter num1 : ");
num1 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter num2 : ");
num2 = Convert.ToDouble(Console.ReadLine());
result = (int)(num1 + num2);
Console.WriteLine("Sum of two numbers = "+result);
```

Output:



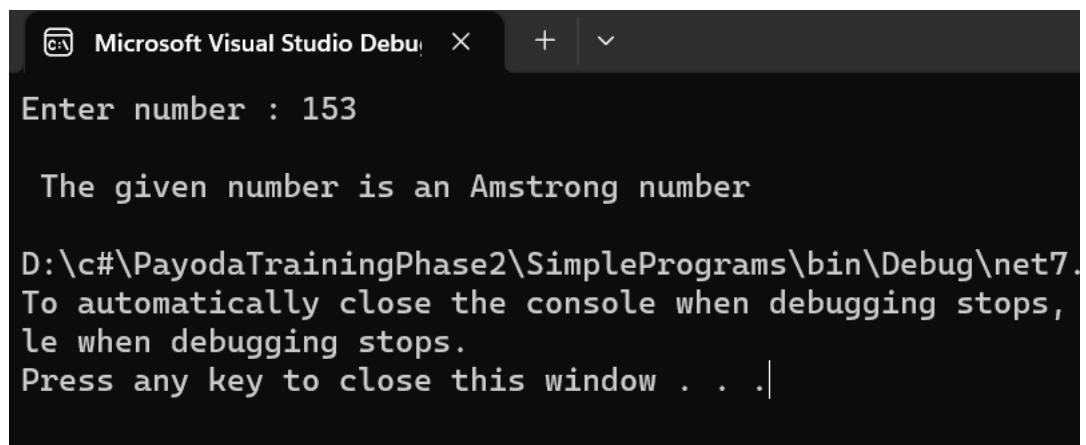
```
Microsoft Visual Studio Debug Console
Enter num1 : 3.3
Enter num2 : 4
Sum of two numbers = 7

D:\c#\PayodaTrainingPhase2\SimplePrograms\bin\Debug\net7.0\SimplePrograms.exe
To automatically close the console when debugging stops, enable
the option 'Close console when debugging stops' in the 'Debug'
tab of the 'Tools' menu.
Press any key to close this window . . .|
```

Amstrong number :

```
int number,temp,sum=0;
Console.Write("Enter number : ");
number=Convert.ToInt32(Console.ReadLine());
temp = number;
while (number > 0)
{
    sum += Convert.ToInt32(Math.Pow((number % 10),3));
    number /= 10;
}
Console.WriteLine("\n The given number is "+((sum==temp)?"an Armstrong
number":"not an Armstrong number"));
```

Output:



```
Microsoft Visual Studio Debug Console
Enter number : 153

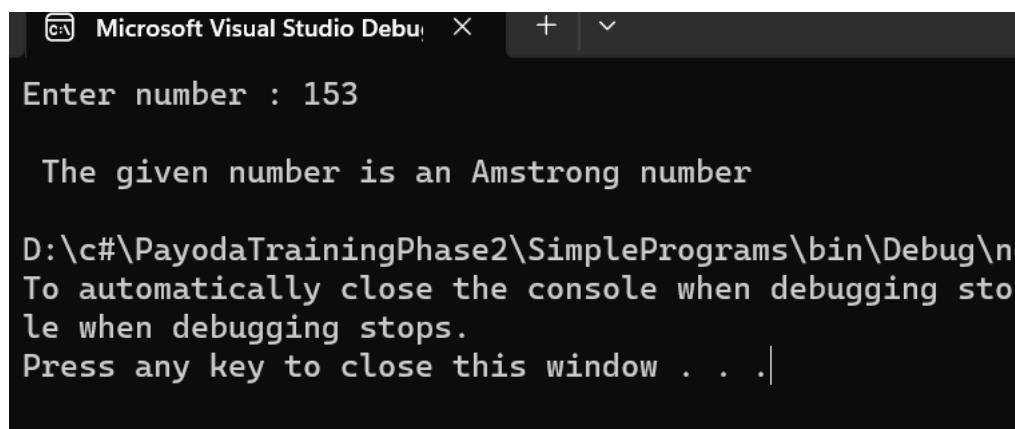
The given number is an Armstrong number

D:\c#\PayodaTrainingPhase2\SimplePrograms\bin\Debug\net7.
To automatically close the console when debugging stops,
le when debugging stops.
Press any key to close this window . . .|
```

Perfect number:

```
int number,sum=1;
Console.Write("Enter number : ");
number = Convert.ToInt32(Console.ReadLine());
for (int i = 2; i <= number/2; i++)
sum += (number % i == 0) ? i : 0;
Console.WriteLine((sum==number)?"{0} is a perfect number": "{0} is not a
perfect number", number);
```

Output:



```
Microsoft Visual Studio Debug Console
Enter number : 153

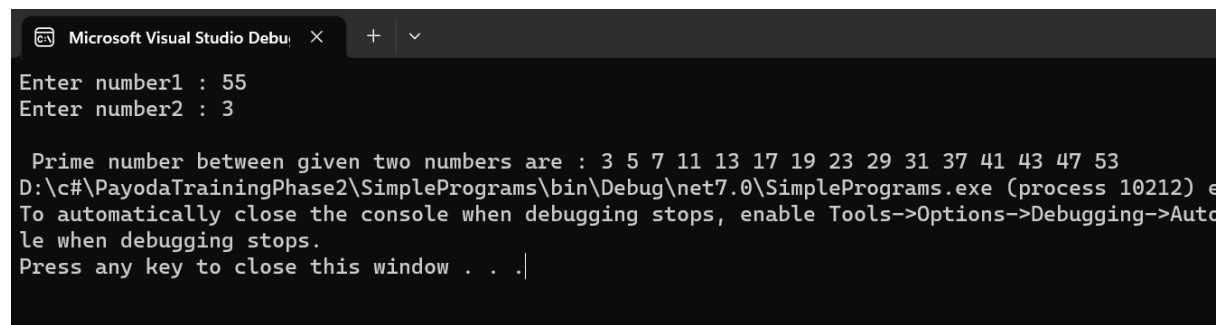
The given number is an Armstrong number

D:\c#\PayodaTrainingPhase2\SimplePrograms\bin\Debug\n
To automatically close the console when debugging sto
le when debugging stops.
Press any key to close this window . . .|
```

Prime numbers between two numbers:

```
int number1,number2,count=0;
Console.Write("Enter number1 : ");
number1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter number2 : ");
number2 = Convert.ToInt32(Console.ReadLine());
int startpoint=(number1 < number2)?number1:number2;
int endpoint = (number1 > number2) ? number1 : number2;
Console.Write("\n Prime number between given two numbers are : ");
for (int i=startpoint;i<=endpoint;i++)
{
    for(int j=1;j<=i/2;j++)
        count += (i % j == 0) ? 1 : 0;
    if(count==1)
        Console.Write(i+" ");
    count = 0;
}
```

Output:



```
Microsoft Visual Studio Debug Console
Enter number1 : 55
Enter number2 : 3

Prime number between given two numbers are : 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53
D:\c#\PayodaTrainingPhase2\SimplePrograms\bin\Debug\net7.0\SimplePrograms.exe (process 10212) e
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Auto
le when debugging stops.
Press any key to close this window . . .|
```