

Website

<https://www.curiousinfotecholutions.com>

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Java Programming Day-11



Email id:
info@curiousinfotecholutions.com

Syllabus Day-11

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



What are the different ways to create an object in Java?

Real World Example-1

Banking System

Real World Example-2

Educational System



Anonymous object

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Anonymous simply means nameless.
- An object which has no reference is known as an anonymous object.
- It can be used at the time of object creation only.
- If you have to use an object only once, an anonymous object is a good approach.
- For example: `new Main();`//anonymous object



Contd.. *Anonymous object*

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Calling method through a reference:
 - `Main c=new Main();`
 - `c.fact(5);//120`
- Calling method through an anonymous object
 - `new Main().fact(5);//120`

Example of an anonymous object in Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- class Main
- {
- void fact(int n)
- {
- int fact=1;
- for(int i=1;i<=n;i++){
- fact=fact*i;
- }
- System.out.println("factorial is "+fact);
- }

```
public static void main(String args[])  
{  
    //calling method with anonymous object  
    new Main().fact(5);  
}
```



<https://www.curiousinfotecholutions.com>

```
class Main
```

```
{
```

```
    int n;
```

```
    Main(int N)
```

```
{
```

```
        n = N;
```

```
}
```

```
void fact()
```

```
{
```

```
    int fact=1;
```

```
    for(int i=1;i<=n;i++){
```

```
        fact=fact*i;
```

```
    }
```

```
    System.out.println("factorial is "+fact);
```

```
}
```

Example of an anonymous object in Java using Constructor

Protected By PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



```
public static void main(String args[])
{
    //calling method with anonymous object
    new Main(5).fact();
}
}
```



Real World Example-1

Java Program to demonstrate the working of a banking-system

Method	Date	AccNO	AccName	Deposited	Withdraw	Balance
AccOpen	16/2/2022	101	Ram	500		500
AccOpen	16/2/2022	102	Shayam	1000		1000
Deposited	17/2/2022	102	Shayam	700		1700
Deposited	17/2/2022	101	Ram	2000		2500
Withdraw	18/2/2022	101	Ram		700	1800
BalRam						1800
CityBankBal						3500
AccClose	18/2/2022	102	Shayam		1700	0
CityBanBal	18/2/2022					1800



Real World Example

Java Program to demonstrate the working of a banking-system

Class Name	Account → Main
Class Data Members	MethodName, TranxDate Acc_no, Account Name depAmt, WithAmt, BalCAmt, BalBAmt
Class Member Functions	1. Open Account, 2. Close Account 3. Display Account Detail Check Balance: 4. CustAmt, 5. BankAmt 6. Amt Deposited 7. Amt Withdraw 8. Exit





- Create Class
 - Create Data Member
 - Create Member Function
 - Open Account
 - Display Account Details

<https://www.curiousinfotecholutions.com>



```
class Main
{
    int AccNo;
    String AccName;
    float AccAmt;
    void openAccount()
    {}
    void displayAccountDetails()
    {}
}
```



```
class Main
{
    int AccNo;
    String AccName;
    float AccAmt;
    void openAccount()
    {}
    void displayAccountDetails()
    {}
}
```



```
void openAccount
(int tAN,String tAName, float tAmt)
{
    AccNo = tAN;
    AccName = tAName;
    AccAmt = tAmt;
}
```



```
class Main
```

```
{  
    int AccNo;  
    String AccName;  
    float AccAmt;  
    void openAccount()  
    {}  
    void displayAccDetails()  
    {}  
}
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



```
void openAccount  
(int tAN, String tAName, float tAmt)
```

```
{  
    AccNo = tAN;  
    AccName = tAName;  
    AccAmt = tAmt;  
}
```

```
void displayAccDetails()
```

```
{  
    System.out.println("AccNo: "+AccNo);  
    System.out.println("AccName: "+AccName);  
    System.out.println("AccAmt: "+AccAmt);  
}
```



Stage - 2

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- ✓ Create Class
- ✓ Create Data Member
- ✓ Create Member Function
 - ✓ Open Account
 - ✓ Display Account Details
- Create Object-1
- Create Object-2

Create Object-1

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- `public static void main (String[] args) {`
- `Main Ram = new Main();`
- `Ram.openAccount(1, "Ram", 500);`
- `Ram.displayAccDetails();`
- `}`

AccNo: 1
AccName: Ram
AccAmt: 500.0

Create Object-2

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- public static void main (String[] args) {
 - Main **Ram** = new Main();
 - Ram.openAccount(1, "Ram", 500);
 - Ram.displayAccDetails();
 - Main **Shayam** = new Main();
 - Shayam.openAccount(2, "Shayam", 1000);
 - Shayam.displayAccDetails();
- }

AccNo: 1
AccName: **Ram**
AccAmt: 500.0
AccNo: 2
AccName: **Shayam**
AccAmt: 1000.0

Stage - 3

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- ✓ Create Class
- ✓ Create Data Member
- ✓ Create Member Function
 - ✓ Open Account
 - ✓ Display Account Details
- ✓ Create Object-1
- ✓ Create Object-2
- Create Member Function
 - Check Balance



Create Member Function - Check Balance



```
• void checkBalance()  
• {  
•     System.out.println("Balance Amt is: "+AccAmt);  
• }
```

```
• Ram.checkBalance();  
• Shayam.checkBalance();
```

Balance Amt is: 500.0
Balance Amt is: 1000.0

AccName	Deposited	Withdraw	Balance
Ram	500		500
Shayam	1000		1000

Stage - 4

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- ✓ Create Class
- ✓ Create Data Member
- ✓ Create Member Function
 - ✓ Open Account
 - ✓ Display Account Details
- ✓ Create Object-1
- ✓ Create Object-2
- Create Member Function
 - ✓ Check Balance
 - Amt Deposited



Create Member Function – Amt Deposited



- void amtDeposited(int tAmtDep)
- {
- AccAmt = AccAmt + tAmtDep;
- }

Balance Amt is: 900.0
Balance Amt is: 1700.0

AccName	Deposited	Withdraw	Balance
Ram	500+400		900
Shayam	1000+700		17000

- //Amt Deposit
- Ram.amtDeposited(400);
- Shayam.amtDeposited(700);
- Ram.checkBalance();
- Shayam.checkBalance();



Stage - 5

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- ✓ Create Class
- ✓ Create Data Member
- ✓ Create Member Function
 - ✓ Open Account
 - ✓ Display Account Details
- ✓ Create Object-1
- ✓ Create Object-2
- Create Member Function
 - ✓ Check Balance
 - ✓ Amt Deposited
 - Amt Withdraw



Create Member Function – Amt Deposited



- void amtWithdraw(int tAmtWithdraw)
- {
- AccAmt = AccAmt - tAmtWithdraw;
- }

Balance Amt is: 500.0
Balance Amt is: 1000.0

AccName	Deposited	Withdraw	Balance
Ram	500+400	500	500
Shayam	1000+700	700	1000

- //Amt amtWithdraw
- Ram.amtWithdraw(400);
- Shayam.amtWithdraw(700);
- Ram.checkBalance();
- Shayam.checkBalance();



```
class Main
{
```

```
    int AccNo;
    String AccName;
    float AccAmt;
```

```
    void openAccount (int tAN,String tAName, float tAmt)
    {
        AccNo = tAN;
        AccName = tAName;
        AccAmt = tAmt;
    }
```

```
    void displayAccDetails()
    {
        System.out.println("AccNo: "+AccNo);
        System.out.println("AccName: "+AccName);
        System.out.println("AccAmt: "+AccAmt);
    }
```

```
    void checkBalance()
    {
        System.out.println("Balance Amt is: "+AccAmt);
    }
```

```
    void amtWithdraw(int tAmtWithdraw)
    {
        AccAmt = AccAmt - tAmtWithdraw;
    }
```

```
    void amtWithdraw(int tAmtWithdraw)
    {
        AccAmt = AccAmt - tAmtWithdraw;
    }
```



```
public static void main (String[] args) {
    //Object Created
    Main Ram = new Main();
    Ram.openAccount(1, "Ram", 500);
    Ram.displayAccDetails();
    Main Shayam = new Main();
    Shayam.openAccount(1, "Shayam", 1000);
    Shayam.displayAccDetails();
```

```
    //Account Balance Checking
    Ram.checkBalance();
    Shayam.checkBalance();
```

```
    //Amt Deposit
    Ram.amtDeposit(400);
    Shayam.amtDeposit(700);
    Ram.checkBalance();
    Shayam.checkBalance();
```

```
    //Amt amtWithdraw
    Ram.amtWithdraw(400);
    Shayam.amtWithdraw(700);
    Ram.checkBalance();
    Shayam.checkBalance();
}
```

Real World Example-2

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Java Program to demonstrate the
working of an educational institution

PID	PType	PDept	PName	PMobile	PCity
101	Faculty	CS	Vijay	9821345678	Pune
102	Staff	HR	Ashok	9833455667	Bhopal
103	Student	CS	Rajesh	9877664433	Mumbai
104	Non-Teaching	EC	Mahesh	9556611223	Nagpur

Real World Example-2

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Java Program to demonstrate the working of an educational institution



Class Name	Person → Main
Class Data Members	PID, PType, Pdept, PName PMobile, PCity
Class Member Functions	1. Add Person
	2. Delete Person
	3. Search Person
	4. Sort Person
	5. List Person
	6. Exit

