**Installing Node.js**

Step-1: Download Node.js from https://nodejs.org/en/download

step-2: Run the .msi installer for Node.

step-3:To verify if the installation was successful, enter the following command in the command prompt.

node –v

step-4: Type the following command in the terminal window (Command Prompt) to install TypeScript.

npm install -g typescript​

**Steps:**

Step-1: Open Visual Studio

Step-2: Select text file (ctl+N)

Step-3: save as filename.ts in your local system in your folder.

filename.ts

Step-4: compile

C:\Users\t-Ravindra1\TypeScriptApps> tsc filename.ts

Note: filename.js will be automatically created after compilation .ts file.

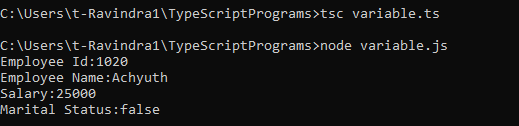
Step-5: Run (Execution)

C:\Users\t-Ravindra1\TypeScriptApps> node filename.js

**Variables:**

|  |
| --- |
| **var employeeId:number=1020;**  **var employeeName:string="Achyuth";**  **var salary:number=25000.00;**  **var maritalStatus:boolean=false;**  **console.log("Employee Id:"+employeeId);**  **console.log("Employee Name:"+employeeName);**  **console.log("Salary:"+salary);**  **console.log("Marital Status:"+maritalStatus);** |

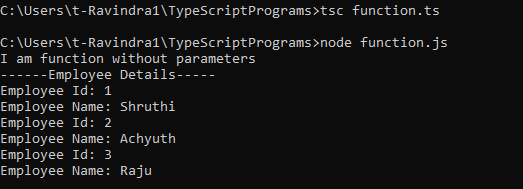
Output:



**Function:**

|  |
| --- |
| **function employeeInfoOne(){**  **console.log("I am function without parameters");**  **}**  **function employeeInfoTwo(employeeId:number,employeeName:string){**    **console.log("Employee Id: "+employeeId);**  **console.log("Employee Name: "+employeeName);**  **}**  **employeeInfoOne();**  **console.log("------Employee Details-----");**  **employeeInfoTwo(1,"Shruthi");**  **employeeInfoTwo(2,"Achyuth");**  **employeeInfoTwo(3,"Raju");** |

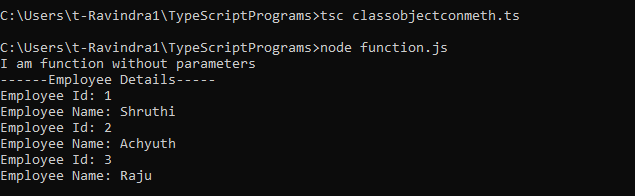
Output:



**Class, object, constructor, method:**

|  |
| --- |
| **class Employee{**  **employeeId:number;**  **employeeName:string;**  **salary:number;**  **constructor(employeeId:number,employeeName:string,salary:number){**  **this.employeeId=employeeId;**  **this.employeeName=employeeName;**  **this.salary=salary;**  **}**  **employeeInfo():void{**  **console.log("---Employee Details----");**  **console.log("Employee Id: "+this.employeeId);**  **console.log("Employee Name: "+this.employeeName);**  **console.log("Salary: "+this.salary);**  **}**  **}**  **var empObject=new Employee(1,"Achyuth",25000.00);**  **empObject.employeeInfo();** |

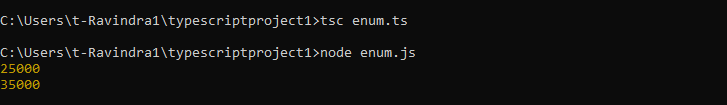
Output:



**Enum:**

|  |
| --- |
| enum Bank {  SavingAccount,  CurrentAccount,  }  interface SavingAccount {  balance:number;    }  interface CurrentAccount {  balance:number;    }  let s=<SavingAccount>{};  s.balance=25000.00;  console.log(s.balance);  let c=<CurrentAccount>{};  c.balance=35000.00;  console.log(c.balance); |

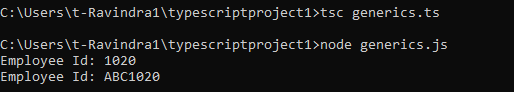
Output:



**Generics:**

|  |
| --- |
| class Employee<Ravi>{  employeeId:Ravi;  constructor(employeeId:Ravi){  this.employeeId=employeeId;  }  employeeInfo():void{  console.log("Employee Id: "+this.employeeId);  }  }  var empObjectNumber=new Employee(1020);  empObjectNumber.employeeInfo();  var empObjectString=new Employee("ABC1020");  empObjectString.employeeInfo(); |

Output:



Modules:

employeeExport.ts

|  |
| --- |
| **export class Employee{**  **employeeId:number;**  **employeeName:string;**  **constructor(employeeId:number,employeeName:string){**  **this.employeeId=employeeId;**  **this.employeeName=employeeName;**  **}**  **employeeInfo():void{**  **console.log("Employee Id: "+this.employeeId);**  **console.log("Employee Name: "+this.employeeName);**  **}**  **}** |

employeeImport.ts

|  |
| --- |
| **import \* as empimp from "./employeeExport"**  **var empObject=new empimp.Employee(1020,"Shruthi");**  **empObject.employeeInfo();** |

Output:

