B.Sc. IN

COMPUTER SCIENCE LAB MANUAL

2nd Semester

Prepared By

Pure and Applied Science Dept.

Computer Science

MIDNAPORE CITY COLLEGE

C3P: PROGRAMMING IN JAVA LABORATORY MANUAL
(Course: CC-3)

INSTRUCTIONS TO STUDENTS

- Before entering the lab, the student should carry the following things (MANDATORY)

    1. Identity card issued by the college.
    2. Class notes
    3. Lab observation book
    4. Lab Manual
    5. Lab Record

- Student must sign in and sign out in the register provided when attending the lab session without fail.
- Come to the laboratory in time. Students, who are late more than 10 min., will not be allowed to attend the lab.
- Students need to maintain 80% attendance in lab if not a strict action will be taken.
- All students must follow a Dress Code while in the laboratory.
- Foods, drinks are NOT allowed.
- All bags must be left at the indicated place.
- Refer to the lab staff if you need any help in using the lab.
- Respect the laboratory and its other users.
- Workspace must be kept clean and tidy after experiment is completed.
- Read the Manual carefully before coming to the laboratory and be sure about what you are supposed to do.
- Do the experiments as per the instructions given in the manual.
- Copy all the programs to observation which are taught in class before attending the lab session.
- Students are not supposed to use floppy disks, pen drives without permission of lab- in charge.
- Lab records need to be submitted on or before the date of submission.

Write a program to find the average and sum of the N numbers using command line arguments.

Program:

```
import java.util.Scanner;
public class Exercise12 {
public static void main(String[] args)
{
int i,n=0,s=0;
double avg;
{
System.out.println("Input the 5 numbers : ");
}
for (i=0;i<5;i++)
{
Scanner in = new Scanner(System.in);
n = in.nextInt();
s +=n;
}
avg=s/5;
System.out.println("The sum of 5 no is : "
+s+"\nThe Average is : " +avg);
}
}
```

Write a program to demonstrate type casting.

Program:

```
import java.util.Scanner;
public class typecasting
{
public static void main(String[] args)
{
//Take input from the user
// create an object of Scanner class
Scanner sc = new Scanner(System.in);
// ask users to enter the number
System.out.println("Enter the number: ");
int i=sc.nextInt();
// widening or automatic type conversion
long l = i;
float f = l;
double d= f;
System.out.println("After widening or
automatic type conversion values are: ");
System.out.println("Int value "+i);
System.out.println("Long value "+l);
System.out.println("Float value "+f);
System.out.println("Double value "+d);
}
}
```

Write a program to calculate simple interest and input by the user.

Program:

```
import java.util.Scanner;
public class simpleinterest
{
public static void main(String args[])
{
float p, r, t, sinterest;
Scanner scan = new Scanner(System.in);
System.out.print("Enter the Principal : ");
p = scan.nextFloat();
System.out.print("Enter the Rate of interest : ");
r = scan.nextFloat();
System.out.print("Enter the Time period : ");
t = scan.nextFloat();
scan.close();
sinterest = (p * r * t) / 100;
System.out.print("Simple Interest is: "
+sinterest);
}
}
```

Write a program to test the prime number.

Program:

```
import java.util.Scanner;
public class PrimeNumber
{
public static void main(String
args[])
{
int num,b,c;
Scanner s=new
Scanner(System.in);
System.out.println("Enter A Number");
num =s.nextInt();
b=1;
c=0;
while(b<= num)
{
if((num%b)==0)
c=c+1;
b++;
}
if(c==2)
System.out.println(num +" is a prime number");
else
System.out.println(num +" is not a prime number");
}
}
```

Write a program to create a simple class to find out the area and perimeter of rectangle and box using super and this keyword.

Program:

```
class rect
{
int l,b;
public rect(int l,int b)
{
this.l=l; this.b=b;
}
public int area()
{
return l*b;
}
}
class box extends rect
{
int d;
public box(int l,int b,int d)
{
super(l,b); this.d=d;
```

```
}
public int volume()
{
int vol = area()*d; return vol;
}
}
class cal
{
public static void main(String args[])
{
int vol ,area;
System.out.println("derived object in derived reference");
rect r= new rect(10,20);
area=r.area();
System.out.println("area is "+area+"\n");
System.out.println("base object in base reference");
box b = new box(10,20,30);
vol=b.volume(); area=b.area();
System.out.println("area is "+area);
System.out.println("volume is "+vol+"\n");
System.out.println("derived object in base reference");
rect b1= new box(10,90,70);
area = b1.area();
//vol=b1.volume(); as with refernce of base class we can't call derived's
System.out.println("area is "+area);
```

```
//as super class doesn't knw abt the base class but reference can be
/*System.out.println("base object in derived reference");
box b2=(new rect (10,20));
vol = b2.area();
System.out.println("area is "+area);*/
r=b;
System.out.println(r.area());
System.out.println(b.volume());
}
}
```

Write a program to find the G.C.D of numbers.

Program:

```
import java.util.Scanner;
public class GCD
{
public static void main(String[] args)
{
//Take input from the user
//Create an instance of the Scanner class
Scanner sc = new Scanner(System.in);
System.out.println("Enter the first number: ");
int num1 = sc.nextInt();
System.out.println("Enter the second number: ");
int num2 = sc.nextInt();
int hcf=0;
for(int i = 1; i <= num1 || i <= num2; i++)
{
if( num1%i == 0 && num2%i == 0 )
hcf = i;
}
System.out.println("HCF of given two numbers is :"+hcf);
}
}
```

Write a program to design a class account using the inheritance and static that show all function of bank.

Program:

```
import java.util.*;
class Bank {
static int acc_no = 10001;
float amt;
public void display() {
System.out.println("Account no :" + acc_no);
System.out.println("Current Amount :" + amt);
}
public Bank() {
amt = 1000;
System.out.println("Ur account no is " + acc_no);
acc_no++;
}
```

```
public void getamt() {
System.out.println("Current balance :" + amt);
}
public void withdraw(float x) {
if (amt == 1000 || amt <= x) {
System.out.println("Sorry u can't withdraw");
} else {
amt = amt - x;
System.out.println("amount withdrawn :" + x);
System.out.println("After withdrawl");
getamt();
}
}
public void deposit(float x) {
if (x == 0)
System.out.println("OOPS 0 can't be deposited");
else {
amt += x;
System.out.println("After deposition");
getamt();
}
}
public static void main(String args[]) {
Scanner sc = new Scanner(System.in);
Bank b1 = new Bank();
b1.deposit(0);
b1.withdraw(120);
b1.display();
System.out.println("\n");
Bank b2 = new Bank();
b2.deposit(1000);
b2.withdraw(150);
}
}
```

Write a program to find the factorial of a given number using recursion.

Program:

```
import java.util.Scanner;
public class Factorial
{
public static void main(String[] args)
{
int n, mul;
Scanner s = new
Scanner(System.in);
System.out.print("Enter any
integer:");
n = s.nextInt();
Factorial obj = new Factorial();
```

```
mul = obj.fact(n);
System.out.println("Factorial of "+n+" :"+mul);
}
int fact(int x)
{
if(x > 1)
{
return(x * fact(x - 1));
}
return 1;
}
}
import java.util.Scanner;
public class Factorial
{
public static void main(String[] args)
{
int n, mul;
Scanner s = new
Scanner(System.in);
System.out.print("Enter any
integer:");
n = s.nextInt();
Factorial obj = new Factorial();
mul = obj.fact(n);
```

```
System.out.println("Factorial of
"+n+" :"+mul);
}
int fact(int x)
{
if(x > 1)
{
return(x * fact(x - 1));
}
return 1;
}
}
```

Write a program to design a class using abstract methods and class.

Program:

```
abstract class Animal {
abstract void makeSound();
public void eat() {
System.out.println("I can eat.");
}
}
class Dog extends Animal {
// provide implementation of abstract method
public void makeSound() {
```

```
System.out.println("Bark bark");
}
}
class abstract {
public static void main(String[] args)
{
// create an object of Dog class
Dog d1 = new Dog();
d1.makeSound();
d1.eat();
}
}
```

Write a program to handle the exception using try and multiple catch block.

Program:

```
public class MultipleCatchBlock {
public static void main(String[] args)
{
try{
int a[]=new int[5];
System.out.println(a[10]);
}
catch(ArithmeticException e)
{
```

```
System.out.println("Arithmetic Exception occurs");
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("ArrayIndexOutOfBounds Exception occurs");
}
catch(Exception e)
{
System.out.println("Parent Exception occurs");
}
System.out.println("rest of the code");
}
}
```

Write a program that implements the nested try statements.

Program:

```
 public class NestedTryBlock {
public static void main(String args[])
{
// outer (main) try block
try {
//inner try block 1
try {
// inner try block 2
try {
int arr[] = { 1, 2, 3, 4 };
//printing the array element out of its bounds
System.out.println(arr[10]);
}
// to handles ArithmeticException
catch (ArithmeticException e)
{
System.out.println("Arithmetic exception");
System.out.println(" inner try block 2");
}
}
// to handle ArithmeticException
catch (ArithmeticException e) {
System.out.println("Arithmetic exception");
System.out.println("inner try block 1");
}
}
// to handleArrayIndexOutOfBoundsException
catch(ArrayIndexOutOfBoundsException e4)
{
System.out.print(e4);
System.out.println(" outer (main) try block");
}
catch (Exception e5) {
```

```
System.out.print("Exception");
System.out.println(" handled in main try-block");
}
}
}
```

Write a program that import the user define package and access the member variable of classes that contained by package.

Program:

```
Package learnjava;
public class First
{
public void msg()
{
System.out.println("HELLO");
}
}
package java ;
import learnjava.*;
class second {
public static void main (String args[])
{
First obj=new First();
obj.msg;
}
}
```

Write a program to create a thrad that

implement the runable interface. public class ExampleClass implements

```
Runnable {
public void run() {
System.out.println("Thread has
ended");
}
public static void main(String[] args)
{
ExampleClass ex = new
ExampleClass();
Thread t1= new Thread(ex);
t1.start();
System.out.println("Hi");
}
}
```

Write a program to implement interthread communication.

Program:

```
 class Customer{
int amount=10000;
synchronized void withdraw(int amount){
System.out.println("going to withdraw...");
if(this.amount<amount){
System.out.println("Less balance; waiting for deposit...");
try{
wait();
}
catch(Exception e){}
}
this.amount-=amount;
System.out.println("withdraw completed...");
}
synchronized void deposit(int amount){
System.out.println("going to deposit...");
this.amount+=amount;
System.out.println("deposit completed... ");
notify();
}
}
class interthread{
```

```
public static void main(String args[]){
final Customer c=new Customer();
new Thread(){
public void run(){c.withdraw(15000);}
}.start();
new Thread(){
public void run(){c.deposit(10000);}
}.start();
}
}
```

Write a program to draw a rectangle using AWT canvas.

Program:

```
import java.awt.Frame;
import java.awt.Canvas;
import java.awt.Color;
import java.awt.Graphics;
public class CanvasDemo
{
private CanvasDemo(){
Frame frame = new Frame("AWT Canvas");
frame.add(new AwtCanvas());
frame.setSize(500,400);
frame.setVisible(true);
}
public static void main (String args[]){
new CanvasDemo();
}
class AwtCanvas extends Canvas
{
AwtCanvas(){
setBackground(Color.cyan);
setSize(400,300);
}
public void paint(Graphics g){
g.setColor(Color.MAGENTA);
g.fillRect(10,10,150,100);
}
}
}
```

Write a program to create a menu using the frame.

Program:

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class menu1 extends JFrame
implements ActionListener{
static JMenuBar mb;
static JMenu x,x1;
static JMenuItem m1,m2,m3,s1,s2;
static JFrame f;
static JLabel l;
public static void main(String args[])
{
menu1 m = new menu1();
f = new JFrame("Menu demo");
l = new JLabel("no task");
mb= new JMenuBar();
x=new JMenu("Menu");
x1= new JMenu("amarjeet");
m1 = new JMenuItem("MenuItem1");
m2 = new JMenuItem("MenuItem2");
m3 = new JMenuItem("MenuItem3");
s1 = new JMenuItem("SubMenuItem1");
s2 = new JMenuItem("SubMenuItem2");
m1.addActionListener(m);
m2.addActionListener(m);
m3.addActionListener(m);
s1.addActionListener(m);
s2.addActionListener(m);
x.add(m1);
x.add(m2);
x.add(m3);
x1.add(s1);
x1.add(s2);
x.add(x1);
mb.add(x);
f.setJMenuBar(mb);
f.add(l);
f.setSize(500,500);
f.setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
String s =e.getActionCommand();
l.setText(s+"selected");
}
}
```

GE-2P: INTRODUCTION TO DATABASE SYSTEM LABORATORY MANUAL (Course: GE-2)

Create and use the following database schema to answer the given queries

EMPLOYEE Schema

Field	**Type**	**NULL**	**KEY**	**DEFAULT**
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	YES	FK	NIL

Hire_date	Date	NO	NIL
Dno	Integer	YES FK	NIL
Commission	Decimal(10,2)	YES	NIL
Salary	Decimal(7,2)	NO	NIL

DEPARTMENT Schema

Field	Type	NULL KEY	DEFAULT
Dno	Integer	NO PRI	NUL
Dname	Varchar(50)	YES	NUL
Location	Varchar(50)	YES	New Delhi

Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employeewith the Employee Number appearing first.

2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.

4. Query to display all the data from the Employee Table. Separate each Column by a commaand name the said column as THE_OUTPUT.

5. Query to display the Employee Name and Salary of all the employees earning more than
$2850.

6. Query to display Employee Name and Department Number for the Employee No= 7900.

7. Query to display Employee Name and Salary for all employees whose salary is not in therange of $1500 and $2850.

8. Query to display Employee Name and Department No. of all the employees in Dept 10 andDept 30 in the alphabetical order by name.

9. Query to display Name and Hire Date of every Employee who was hired in 1981.

10. Query to display Name and Job of all employees who don‘t have a current Manager.

11. Query to display the Name, Salary and Commission for all the employees who earn commission. Sort the data in descending order of Salary and Commission.

12. Query to display Name of all the employees where the third letter of their name is ‗A‘.

13. Query to display Name of all employees either have two ‗R‘s or have two ‗A‘s in their name and are either in Dept No = 30 or their Manger‘s Employee No = 7788.

14. Query to display Name, Salary and Commission for all employees whose CommissionAmount is 14 greater than their Salary increased by 5%.

15. Query to display the Current Date.

16. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after sixmonths of employment.

17. Query to display Name and calculate the number of months between today and the date eachemployee was hired.

18. Query to display the following for each employee <E-Name> earns < Salary> monthly butwants < 3 * Current Salary >. Label the Column as Dream Salary.

19. Query to display Name with the 1st letter capitalized and all other letter lower case and
length of their name of all the employees whose name starts with ‗J‘, ‘A‘ and ‗M‘.

20. Query to display Name, Hire Date and Day of the week on which the employee started.

21. Query to display Name, Department Name and Department No for all the employees.

22. Query to display Unique Listing of all Jobs that are in Department # 30.

23. Query to display Name, Dept Name of all employees who have an _ A‘ in their name.

24. Query to display Name, Job, Department No. and Department Name for all the employees working at the Dallas location.

25. Query to display Name and Employee no. Along with their Manger‘s Name and the Manager‘s employee no; along with the Employees‘ Name who do not have a Manager.

26. Query to display Name, Dept No. and Salary of any employee whose department No. and salary matches both the department no. and the salary of any employee who earns a commission.

27. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.

28. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees

29. Query to display the number of employees performing the same Job type functions.

30. Query to display the no. of managers without listing their names.

31. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.

32. Query to display Name and Hire Date for all employees in the same dept. as Blake.

33. Query to display the Employee No. and Name for all employees who earn more than the average salary.

34. Query to display Employee Number and Name for all employees who work in a department
with any employee whose name contains a _T‘.

35. Query to display the names and salaries of all employees who report to King.

36. Query to display the department no, name and job for all employees in the Sales department.

Table Creation

SQL> create table department(Dno number(10), Dname varchar2(20), Location varchar2(20), primary key (Dno));

SQL> create table employee(Eno char(3), Ename varchar2(20), Job_type varchar2(20), Manager char(3), Hire_date date, Dno number(10), Commission decimal(10, 2), Salary decimal(7,2), primary key(Eno), constraint Dno foreign key (Dno) references department (Dno));

Table Description

SQL> desc department

```
 Name                                    Null?    Type
 --------------------------------------- -------- ----------------------------
 DNO                                     NOT NULL NUMBER(10)
 DNAME                                            VARCHAR2(20)
 LOCATION                                         VARCHAR2(20)
```

SQL> desc employee;

```
 Name                                    Null?    Type
 --------------------------------------- -------- ----------------------------
 ENO                                     NOT NULL CHAR(3)
 ENAME                                            VARCHAR2(20)
 JOB_TYPE                                         VARCHAR2(20)
 MANAGER                                          CHAR(3)
 HIRE_DATE                                        DATE
 DNO                                              NUMBER(10)
 COMMISSION                                       NUMBER(10,2)
 SALARY                                           NUMBER(7,2)
```

Insertion of values to Tables

Department Table

```
SQL> insert into department values(10, 'Accounting', 'New York');


1 row created.


SQL> insert into department values(20, 'Research', 'Dallas');


1 row created.


SQL> insert into department values(30, 'Sales', 'Chicago');


1 row created.


SQL> insert into department values(40, 'Operation', 'Boston');


1 row created.


SQL> insert into department values(50, 'Marketing', 'New Delhi');


1 row created.
SQL> select * from department;


       DNO DNAME                LOCATION
---------- -------------------- --------------------
        10 Accounting           New York
```

```
20 Research      Dallas
30 Sales        Chicago
40 Operation     Boston
50 Marketing     New Delhi
```

Employee Table

```
SQL> insert into employee values('736', 'Smith', 'Clerk', '790',
to_date('17/12/1981','dd/mm/yyyy'), 20, 0.00, 1000.00);


1 row created.


SQL> insert into employee values('749', 'Allan', 'Sales_man', '769',
to_date('20/02/1981','dd/mm/yyyy'), 30, 300.00, 2000.00);


1 row created.


SQL> insert into employee values('752', 'Ward', 'Sales_man', '769',
to_date('22/02/1981','dd/mm/yyyy'), 30, 500.00, 1300.00);


1 row created.


SQL> insert into employee values('756', 'Jones', 'Manager', '783',
to_date('02/04/1981','dd/mm/yyyy'), 20, 0.00, 2300.00);


1 row created.


SQL> insert into employee values('765', 'Martin', 'Sales_man', '784',
to_date('22/04/1981','dd/mm/yyyy'), 30, 1400.00, 1250.00);
```

```
1 row created.

SQL> insert into employee values('769', 'Blake', 'Manager', '783',
to_date('01/05/1981','dd/mm/yyyy'), 30, 0.00, 2870.00);

1 row created.

SQL> insert into employee values('778', 'Clark', 'Manager', '783',
to_date('09/06/1981','dd/mm/yyyy'), 10, 0.00, 2900.00);

1 row created.

SQL> insert into employee values('783', 'King', 'President', NULL,
to_date('17/11/1981','dd/mm/yyyy'), 10, 0.00, 2950.00);

1 row created.

SQL> insert into employee values('784', 'Turner', 'Sales_man', '769',
to_date('08/09/1981','dd/mm/yyyy'), 30, 0.00, 1450.00);

1 row created.

SQL> commit;

Commit complete.

SQL> insert into employee values('787', 'Adams', 'Clerk', '778',
to_date('12/01/1983','dd/mm/yyyy'), 20, 0.00, 1150.00);
```

```
1 row created.

SQL> insert into employee values('788', 'Scott', 'Analyst', '756',
to_date('09/12/1982','dd/mm/yyyy'), 20, 0.00, 2850.00);

1 row created.

SQL> insert into employee values('790', 'James', 'Clerk', '769',
to_date('03/12/1981','dd/mm/yyyy'), 30, 0.00, 950.00);

1 row created.

SQL> insert into employee values('792', 'Ford', 'Analyst', '756',
to_date('03/12/1981','dd/mm/yyyy'), 20, 0.00, 2600.00);

1 row created.

SQL> insert into employee values('793', 'Miller', 'Clerk', '788',
to_date('23/01/1982','dd/mm/yyyy'), 40, 0.00, 1300.00);

1 row created.
SQL> select * from employee;

ENO ENAME                JOB_TYPE             MAN HIRE_DATE        DNO
--- -------------------- -------------------- --- --------- ----------
COMMISSION     SALARY
---------- ----------
788 Scott                Analyst              756 09-DEC-82         20
         0       2850
```

```
736 Smith               Clerk               790 17-DEC-81         20

         0       1000


749 Allan               Sales_man           769 20-FEB-81         30

       300       2000



ENO ENAME               JOB_TYPE            MAN HIRE_DATE         DNO
--- -------------------- -------------------- --- --------- ----------
COMMISSION     SALARY
---------- ----------
752 Ward                Sales_man           769 22-FEB-81         30

       500       1300


756 Jones               Manager             783 02-APR-81         20

         0       2300


765 Martin              Sales_man           784 22-APR-81         30

      1400       1250



ENO ENAME               JOB_TYPE            MAN HIRE_DATE         DNO
--- -------------------- -------------------- --- --------- ----------
COMMISSION     SALARY
---------- ----------
769 Blake               Manager             783 01-MAY-81         30
```

```
         0       2870

778 Clark              Manager             783 09-JUN-81         10
         0       2900


783 King               President              17-NOV-81         10
         0       2950



ENO ENAME              JOB_TYPE            MAN HIRE_DATE        DNO
--- -------------------- -------------------- --- --------- ----------
COMMISSION     SALARY
---------- ----------
784 Turner              Sales_man            769 08-SEP-81         30
         0       1450


787 Adams               Clerk                778 12-JAN-83         20
         0       1150


793 Miller              Clerk                788 23-JAN-82         40
         0       1300



ENO ENAME              JOB_TYPE            MAN HIRE_DATE        DNO
--- -------------------- -------------------- --- --------- ----------
COMMISSION     SALARY
---------- ----------
```

```
790 James          Clerk          769 03-DEC-81       30
      0       950


792 Ford           Analyst        756 03-DEC-81       20
     14      2600



14 rows selected.
```

1. **Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.**
   SQL> select Eno, Ename, Job_type, Hire_date from employee;

```
ENO ENAME              JOB_TYPE            HIRE_DATE
--- -------------------- -------------------- ---------
788 Scott              Analyst             09-DEC-82
736 Smith              Clerk               17-DEC-81
749 Allan              Sales_man           20-FEB-81
752 Ward               Sales_man           22-FEB-81
756 Jones              Manager             02-APR-81
765 Martin             Sales_man           22-APR-81
769 Blake              Manager             01-MAY-81
778 Clark              Manager             09-JUN-81
783 King               President           17-NOV-81
784 Turner             Sales_man           08-SEP-81
787 Adams              Clerk               12-JAN-83
790 James              Clerk               03-DEC-81
792 Ford               Analyst             03-DEC-81
793 Miller             Clerk               23-JAN-82
```

2. **Query to display unique Jobs from the Employee Table.**
   SQL> select distinct Job_type from employee;

```
JOB_TYPE
--------------------
Analyst
```

```
Clerk
Manager
President
Sales_man
```

3. **Query to display the Employee Name concatenated by a Job separated by a comma.**

```
SQL> select Ename||', '|| Job_type as Name_Job from employee;

NAME_JOB
-----------------------------------------
Scott, Analyst
Smith, Clerk
Allan, Sales_man
Ward, Sales_man
Jones, Manager
Martin, Sales_man
Blake, Manager
Clark, Manager
King, President
Turner, Sales_man
Adams, Clerk
Miller, Clerk
James, Clerk
Ford, Analyst

14 rows selected.
```

4. **Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.**

```
SQL> select Eno||', '||Ename||', '||Job_type||', '||Manager||', '||Hire_date||',
'||Dno||', '||Commission||', '||Salary from employee ;

ENO||','||ENAME||','||JOB_TYPE||','||MANAGER||','||HIRE_DATE||','||DN
O||','||COM
-------------------------------------------------------------------------------
788, Scott, Analyst, 756, 09-DEC-82, 20, 0, 2850
736, Smith, Clerk, 790, 17-DEC-81, 20, 0, 1000
749, Allan, Sales_man, 769, 20-FEB-81, 30, 300, 2000
```

```
752, Ward, Sales_man, 769, 22-FEB-81, 30, 500, 1300
756, Jones, Manager, 783, 02-APR-81, 20, 0, 2300
765, Martin, Sales_man, 784, 22-APR-81, 30, 1400, 1250
769, Blake, Manager, 783, 01-MAY-81, 30, 0, 2870
778, Clark, Manager, 783, 09-JUN-81, 10, 0, 2900
783, King, President, , 17-NOV-81, 10, 0, 2950
784, Turner, Sales_man, 769, 08-SEP-81, 30, 0, 1450
787, Adams, Clerk, 778, 12-JAN-83, 20, 0, 1150
793, Miller, Clerk, 788, 23-JAN-82, 40, 0, 1300
790, James, Clerk, 769, 03-DEC-81, 30, 0, 950
792, Ford, Analyst, 756, 03-DEC-81, 20, 0, 2600

14 rows selected.
```

5. **Query to display the Employee Name and Salary of all the employees earning more than $2850.**

```
SQL> select Ename, salary from employee where
(salary+commission)>2850;

ENAME                SALARY
-------------------- ----------
Blake                2870
Clark                2900
King                 2950
```

6. **Query to display Employee Name and Department Number for the Employee No= 790.**

```
SQL> select Ename, Dno from employee where Eno='790';

ENAME                  DNO
-------------------- ----------
James                  30
```

7. **Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.**

```
SQL> select Ename, salary from employee where salary not between
1500 and 2850;

ENAME                SALARY
```

```
Smith          1000
Ward           1300
Martin         1250
Blake          2870
Clark          2900
King           2950
Turner         1450
Adams          1150
Miller         1300
James           950

10 rows selected.
```

8. **Query to display Employee Name and Department No. Of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.**

```
SQL> select Ename, Dno from employee where (Dno=10 or Dno=30)
order by (Ename);

ENAME                DNO
-------------------- ----------
Allan                30
Blake                30
Clark                10
James                30
King                 10
Martin               30
Turner               30
Ward                 30

8 rows selected.
```

9. **Query to display Name and Hire Date of every Employee who was hired in 1981.**

```
SQL> select Ename, Hire_date from employee where to_char(Hire_date,
'yyyy')='1981';

ENAME           HIRE_DATE
```

```
--------------------
Smith           17-DEC-81
Allan          20-FEB-81
Ward            22-FEB-81
Jones          02-APR-81
Martin          22-APR-81
Blake           01-MAY-81
Clark          09-JUN-81
King            17-NOV-81
Turner          08-SEP-81
James           03-DEC-81
Ford           03-DEC-81

11 rows selected.
```

10.Query to display Name and Job of all employees who don't have a current Manager.

```
SQL> select Ename, Job_type from employee where Manager is NULL;

ENAME              JOB_TYPE
-------------------- --------------------
King               President
```

11.Query to display the Name, Salary and Commission for all the employees who earn commission. Sort the data in descending order of Salary and Commission.

```
SQL> select Ename, Salary, Commission from employee where
(Commission > 0.00) order by (Salary) desc;

ENAME                SALARY COMMISSION
-------------------- ---------- ----------
Allan                  2000       300
Ward                    1300       500
Martin                  1250      1400
```

12.Query to display Name of all the employees where the third letter of their name is 'a'.

```
SQL> select Ename from employee where Ename like '__a%';

ENAME
--------------------
Blake
Clark
Adams
```

13. Query to display Name of all employees either have two 'r's or have two 'a's in their name and are either in Dept No = 30 or their Manger's Employee No = 778.

```
SQL> select Ename, Dno, Manager from employee where Ename like
'%a%a' or Ename like '%r%r' and Dno=30 or Manager='778';

ENAME              DNO MAN
-------------------- ---------- ---
Turner                30 769
Adams                 20 778
```

14. Query to display Name, Salary and Commission for all employees whose Commission Amount is greater than their Salary increased by 5%.

```
SQL> select Ename, Salary, Commission from employee where
Commission > (Salary + Salary * 0.05);

ENAME              SALARY COMMISSION
-------------------- ---------- ----------
Martin                1250       1400
```

15. Query to display the Current Date.

```
SQL> select Sysdate from Dual;

SYSDATE
---------
25-JUN-23
```

16.Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.

```
SQL> SELECT Ename,
Hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(Hire_date, 6),
'MONDAY'),'fmDay, " the " Ddspth " of " Month, YYYY') as "REVIEW"
FROM employee;

ENAME                HIRE_DATE
-------------------- ---------
REVIEW
--------------------------------------------------
Scott                09-DEC-82
Monday,  the  Thirteenth  of  June, 1983

Smith                17-DEC-81
Monday,  the  Twenty-First  of  June, 1982

Allan                20-FEB-81
Monday,  the  Twenty-Fourth  of  August, 1981


ENAME                HIRE_DATE
-------------------- ---------
REVIEW
--------------------------------------------------
Ward                 22-FEB-81
Monday,  the  Twenty-Fourth  of  August, 1981

Jones                02-APR-81
Monday,  the  Fifth  of  October, 1981

Martin               22-APR-81
Monday,  the  Twenty-Sixth  of  October, 1981


ENAME                HIRE_DATE
-------------------- ---------
```

```
REVIEW
----------------------------------------------------
Blake            01-MAY-81
Monday,  the  Second  of  November, 1981

Clark            09-JUN-81
Monday,  the  Fourteenth  of  December, 1981

King             17-NOV-81
Monday,  the  Twenty-Fourth  of  May, 1982


ENAME              HIRE_DATE
-------------------- ---------
REVIEW
----------------------------------------------------
Turner           08-SEP-81
Monday,  the  Fifteenth  of  March, 1982

Adams             12-JAN-83
Monday,  the  Eighteenth  of  July, 1983

Miller           23-JAN-82
Monday,  the  Twenty-Sixth  of  July, 1982


ENAME              HIRE_DATE
-------------------- ---------
REVIEW
----------------------------------------------------
James            03-DEC-81
Monday,  the  Seventh  of  June, 1982

Ford             03-DEC-81
Monday,  the  Seventh  of  June, 1982


14 rows selected.
```

17. Query to display Name and calculate the number of months between today and the date each employee was hired.

```
SQL> select Ename, Round(Months_Between(sysdate,Hire_date)) as
"Months_Worked" from employee;

ENAME              Months_Worked
-------------------- -------------
Scott                    487
Smith                     498
Allan                    508
Ward                      508
Jones                    507
Martin                    506
Blake                    506
Clark                    505
King                     499
Turner                    502
Adams                      485
Miller                   497
James                     499
Ford                     499

14 rows selected.
```

18. Query to display the following for each employee:- <E-Name> earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.

```
SQL> select Ename||' earns $'||Salary||' monthly but wants $'||salary*3
"Dream Salary" from employee;

Dream Salary
--------------------------------------------------------------------------------
Scott earns $2850 monthly but wants $8550
Smith earns $1000 monthly but wants $3000
Allan earns $2000 monthly but wants $6000
Ward earns $1300 monthly but wants $3900
Jones earns $2300 monthly but wants $6900
```

```
Martin earns $1250 monthly but wants $3750
Blake earns $2870 monthly but wants $8610
Clark earns $2900 monthly but wants $8700
King earns $2950 monthly but wants $8850
Turner earns $1450 monthly but wants $4350
Adams earns $1150 monthly but wants $3450
Miller earns $1300 monthly but wants $3900
James earns $950 monthly but wants $2850
Ford earns $2600 monthly but wants $7800

14 rows selected.
```

19. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with ‘J’, ’A’ and ‘M’.

```
SQL> select initcap(Ename) "Name", length(Ename) "Length of Name"
from employee where Ename like 'J%' or Ename like 'A%'
 or Ename like 'M%' order by Ename;

Name                 Length of Name
-------------------- --------------
Adams                             5
Allan                          5
James                           5
Jones                          5
Martin                          6
Miller                         6

6 rows selected.
```

20. Query to display Name, Hire Date and Day of the week on which the employee started.

```
SQL> SELECT  Ename, Hire_date, TO_CHAR(Hire_date,'DAY') AS
DAY FROM employee ORDER BY Hire_date, DAY;

ENAME                HIRE_DATE DAY
-------------------- --------- ---------
```

```
Allan          20-FEB-81 FRIDAY
Ward            22-FEB-81 SUNDAY
Jones          02-APR-81 THURSDAY
Martin          22-APR-81 WEDNESDAY
Blake           01-MAY-81 FRIDAY
Clark          09-JUN-81 TUESDAY
Turner          08-SEP-81 TUESDAY
King            17-NOV-81 TUESDAY
James           03-DEC-81 THURSDAY
Ford           03-DEC-81 THURSDAY
Smith           17-DEC-81 THURSDAY
Miller         23-JAN-82 SATURDAY
Scott          09-DEC-82 THURSDAY
Adams            12-JAN-83 WEDNESDAY

14 rows selected.
```

21.Query to display Name, Department Name and Department No for all the employees.

```
SQL> select employee.Ename,department.Dname,employee.Dno from
employee, department where employee.Dno=department.Dno;

ENAME              DNAME                   DNO
-------------------- -------------------- ----------
Scott          Research              20
Smith           Research              20
Allan          Sales               30
Ward            Sales               30
Jones          Research              20
Martin          Sales               30
Blake           Sales               30
Clark          Accounting              10
King           Accounting              10
Turner          Sales               30
Adams            Research              20
Miller         Operation             40
James           Sales               30
Ford           Research              20
```

```
14 rows selected.
```

22.Query to display Unique Listing of all Jobs that are in Department # 30.

```
SQL> select distinct Job_type from employee where Dno=30;

JOB_TYPE
--------------------
Manager
Clerk
Sales_man
```

23.Query to display Name, Dept Name of all employees who have an 'a' in their name.

```
SQL> select employee.Ename,department.Dname from
employee,department where employee.Ename like '%a%' and
employee.Dno=department.Dno;

ENAME              DNAME
-------------------- --------------------
Allan              Sales
Ward               Sales
Martin             Sales
Blake              Sales
Clark              Accounting
Adams              Research
James              Sales

7 rows selected.
```

24.Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.

```
SQL> select employee.Ename, employee.Job_type, employee.Dno,
department.Dname from employee,department where
employee.Dno=department.Dno and department.Location='Dallas';
```

```
ENAME              JOB_TYPE              DNO DNAME
-------------------- -------------------- ---------- --------------------
Scott              Analyst               20 Research
Smith              Clerk                 20 Research
Jones              Manager               20 Research
Adams              Clerk                 20 Research
Ford               Analyst               20 Research
```

25.Query to display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees' Name who do not have a Manager.

SQL> select e.Ename,e.Eno,d.Ename,d.Eno from employee e left outer join employee d ON e.Eno=d.Manager;

```
ENAME              ENO ENAME              ENO
-------------------- --- -------------------- ---
Jones              756 Scott              788
James              790 Smith              736
Blake              769 Allan              749
Blake              769 Ward               752
King               783 Jones              756
Turner             784 Martin             765
King               783 Blake              769
King               783 Clark              778
Blake              769 Turner             784
Clark              778 Adams              787
Scott              788 Miller             793

ENAME              ENO ENAME              ENO
-------------------- --- -------------------- ---
Blake              769 James              790
Jones              756 Ford               792
Miller             793
Ward               752
Martin             765
Smith              736
Allan              749
```

```
Ford           792
Adams            787

20 rows selected.
```

26.Query to display Name, Dept No. And Salary of any employee whose department No. And salary matches both the department no. And the salary of any employee who earns a commission.

```
SQL> select Ename,Dno,Salary from employee where (Dno,Salary) in
(select Dno,Salary from employee where Commission>0);

ENAME                 DNO     SALARY
-------------------- ---------- ----------
Allan                  30       2000
Ward                   30       1300
Martin                 30       1250
```

27.Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.

```
SQL> select Ename, RPAD('*', Salary/100) as Salary_Representation
from employee;

ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Scott
*


Smith
*


Allan
*



ENAME
--------------------
```

```
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Ward
*

Jones
*

Martin
*


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Blake
*

Clark
*

King
*


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Turner
*

Adams
*

Miller
*
```

```
ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
James
*

Ford
*


14 rows selected.

SQL> select Ename, RPAD('*', Salary/100) as Salary_Representation
from employee;

ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Scott
*

Smith
*

Allan
*


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Ward
*

Jones
```

```
*

Martin
*


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Blake
*

Clark
*

King
*


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Turner
*

Adams
*

Miller
*


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
James
```

```
*
Ford
*



14 rows selected.

SQL> SELECT Ename, RPAD('*', CEIL(Salary/100), '*') as
Salary_Representation FROM employee;

ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Scott
*****************************

Smith
**********

Allan
********************


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Ward
*************

Jones
***********************

Martin
*************
```

```
ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Blake
*****************************

Clark
*****************************

King
******************************


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Turner
***************

Adams
************

Miller
*************


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
James
**********

Ford
**************************
```

```
14 rows selected.


SQL> SELECT Ename, RPAD('*', (Salary/100), '*') as
Salary_Representation FROM employee;

ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Scott
****************************

Smith
**********

Allan
********************


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Ward
*************

Jones
***********************

Martin
************


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
```

```
Blake
****************************

Clark
*****************************

King
*****************************


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
Turner
**************

Adams
***********

Miller
*************


ENAME
--------------------
SALARY_REPRESENTATION
--------------------------------------------------------------------------------
James
*********

Ford
**************************


14 rows selected.
```

28. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees.

```
SQL> select MAX(Salary),MIN(Salary),SUM(Salary),AVG(Salary) from
employee;

MAX(SALARY) MIN(SALARY) SUM(SALARY) AVG(SALARY)
----------- ----------- ----------- -----------
      2950        950       26870  1919.28571
```

29.Query to display the number of employees performing the same Job type functions.

```
SQL> select Job_type,COUNT(*) from employee group by Job_type;

JOB_TYPE             COUNT(*)
-------------------- ----------
Analyst                 2
Clerk                   4
Manager                   3
President                1
Sales_man                  4
```

30.Query to display the no. Of managers without listing their names.

```
SQL> select COUNT(DISTINCT Manager) from employee;

COUNT(DISTINCTMANAGER)
----------------------
                    7
```

31.Query to display the Department Name, Location Name, No. Of Employees and the average salary for all employees in that department.

```
SQL> SELECT d.Dname, d.Location, COUNT(*), AVG(e.Salary) from
Department d JOIN Employee e ON d.Dno = e.Dno GROUP BY
d.Dname, d.Location;

DNAME              LOCATION    COUNT(*) AVG(E.SALARY)
-------------------- -------------------- ---------- -------------
Research           Dallas                   5         1980
```

```
Sales          Chicago          6    1636.66667
Accounting        New York      2       2925
Operation        Boston          1      1300
```

32. Query to display Name and Hire Date for all employees in the same dept. As Blake.

```
SQL> select Ename,Hire_date from employee where Dno=(select Dno
from employee where Ename='Blake');

ENAME              HIRE_DATE
-------------------- ---------
Allan            20-FEB-81
Ward              22-FEB-81
Martin            22-APR-81
Blake             01-MAY-81
Turner            08-SEP-81
James             03-DEC-81

6 rows selected.
```

33. Query to display the Employee No. And Name for all employees who earn more than the average salary.

```
SQL> select Eno,Ename from employee where Salary > (Select
AVG(Salary) from employee);

ENO ENAME
--- --------------------
788 Scott
749 Allan
756 Jones
769 Blake
778 Clark
783 King
792 Ford

7 rows selected.
```

34. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 't'.

```
SQL> select e.Eno,e.Ename from employee e ,employee d where
e.Manager=d.Eno and d.Ename like '%t%';

ENO ENAME
--- --------------------
793 Miller
```

35. Query to display the names and salaries of all employees who report to King.

```
SQL> select Ename,Salary from employee where Manager=(select Eno
from employee where Ename='King');

ENAME                SALARY
-------------------- ----------
Jones                 2300
Blake                 2870
Clark                 2900
```

36. Query to display the department no, name and job for all employees in the Sales department.

```
SQL> select e.Dno,e.Ename,e.Job_type from employee e,department d
where d.Dno=e.Dno and d.Dname='Sales';

       DNO ENAME                JOB_TYPE
---------- -------------------- --------------------
        30 Allan                Sales_man
        30 Ward                 Sales_man
        30 Martin               Sales_man
        30 Blake                Manager
        30 Turner               Sales_man
        30 James                Clerk

6 rows selected.
```