MS-11L

# Karnataka State Open University

## M.Sc (IT) I Semester Practical List [New Scheme]

Time : 3hrs C-Programming Lab Max Marks : 100

1. PROGRAM to find the root of quadratic equation using nested IF.
2. Given two numbers PROGRAM to perform arithmetic operations using Switch statements.
3. PROGRAM to generate Fibonacci series up to N numbers using Do-While loop.
4. PROGRAM to find reverse of the given number, sum & count the number of digits & Check whether the given no is Palindrome or not using While – do loop.
5. PROGRAM to generate Prime numbers using For loop.
6. PROGRAM to search an element using linear technique.
7. PROGRAM to add & subtract two M x N Matrices.
8. PROGRAM to multiply M x N Matrices.
9. PROGRAM to Transpose M x N Matrix using function.
10. PROGRAM to find the factorial of a number using recursion.
11. PROGRAM to swap numbers using function & pointers.
12. PROGRAM to insert a sub-string into a string.
13. PROGRAM to read &write information of an employee using structure.
14. PROGRAM to find largest & smallest of n numbers.
15. PROGRAM to sort numbers using selection sort.
16. PROGRAM to construct Pyramid of digits.
17. PROGRAM to generate Pascal triangle.
18. PROGRAM to check whether the given number is a Fibonacci term or not.
19. PROGRAM to check whether the given number is a factorial of a number or not.
20. PROGRAM to find the second largest of N numbers.

MS-12L

# Karnataka State Open University

M.Sc (IT) I Semester Practical List [New Scheme]

Time : 3hrs Analysis and Design of Algorithms Max Marks : 100

## Write Programs using C and Execute

1. Implement Selection sort technique and find time and space complexity.
2. Implement Insertion sort technique and find time and space complexity.
3. Implement Merge sort technique and find time and space complexity.
4. Implement Hashing technique to determine whether an arbitrary element is present or not.
5. Write a program to find minimal spanning tree traversal considering your own tree.
6. Write a program to find the product of two numbers using Halving and Doubling method.
7. Implement Knapsack problem.
8. Implement Heap sort technique.
9. Implement backtracking algorithm.
10. Implement Job scheduling method.
11. Implement Cassette filling method.
12. Sort number in ascending order using Bubble sort.

MS-14L

# Karnataka State Open University

M.Sc (IT) I Semester Practical List [New Scheme]

Time : 3hrs Advanced RDBMS (Lab) Max Marks : 100

1. Create a table 'STUDENT' to store the details of marks of a student.

| Field | Type | Width | constraint |
|---|---|---|---|
| Student_ID | Numeric | 5 | Unique |
| Name | Text | 20 | |
| class | Numeric | 2 | |
| English | Numeric | 3 | |
| Hindi | Numeric | 3 | |
| Maths | Numeric | 3 | |
| Science | Numeric | 3 | |
| Social_Science | Numeric | 3 | |

Create a table 'Transaction' to have the following fields.

| Field | Type | Width | constraint |
|---|---|---|---|
| Trans_No | Numeric | 5 | Unique |
| Item_No | Numeric | 5 | |
| Item_Name | Text | 25 | |
| Trans_Date | date | | |
| Quantity | Numeric | 5 | |

After creating the tables, do the following:

a. Set field properties of each field.

b. Modify fields in the table.

c. Modify the table 'STUDENT' to include the following fields:

| Field | Type | Width | constraint |
|---|---|---|---|
| Total | Numeric | 4 | |
| Average | Numeric | 5 | 2 decimal places |
| Result | Text | 10 | |

d. Apply necessary validation rules to each field.

e. Rename the field 'Total' by 'Aggregate'.

f. Delete field 'Result'.

g. Add records.

2. For the tables created in above problem, do the following:

a. Apply filters to list students with marks greater then 60.
b. Apply filters to get transactions for a date.
c. Sort students by name.
d. Sort transactions by date.
e. Create queries to list students with Aggregate <300 and Aggregate >= 250.
f. Total transaction quantity for a given date.

3. Using the tables created in problem number 1, do the following:

a. Create forms to view data.
b. Add, Delete and Save records through the forms created.
c. Change the structure of the above forms in design view.

4. Using the tables created in problem number 1, and/ or related queries, generate the following reports:

a. List of students with marks >60 in English.
b. List of students whose Average is >80.
c. List of Items for a given Transaction date.
d. Day-wise transactions for each month under the month's heading showing total transaction at the end.

5. Create a table "EMPLOYEE" to store the details with following fields (at least 10 records)

| Field | Type | Width | Constraints |
|---|---|---|---|
| Employee id | Number | 5 | Primary key |
| Name | String | 20 | |
| Address | Text | 50 | |
| Basic salary | Number | 6 | Validate for not accepting more than 2 decimal places |
| Net salary | Number | 6 | |

6. Using above table Generate the following query with reports
a) By Employee No
b) By salary wise
c) list the employees, whose basic salary is > 10000

7. Consider the following tables GAMES and PLAYER. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii)

**GAMES**

| GCode | GameName | Number | PrizeMoney | Date |
|---|---|---|---|---|
| 101 | CaromBoard | 2 | 5000 | 23-jan-2004 |
| 102 | Badminton | 2 | 12000 | 12-dec-2003 |
| 103 | TableTennis | 4 | 8000 | 14-feb-2004 |
| 104 | Chess | 2 | 9000 | 01-jan-2004 |
| 105 | LawnTennis | 4 | 25000 | 19-mar-2004 |

**PLAYER**

| PCode | Name | GCode |
|---|---|---|
| 1 | Arjun | 101 |
| 2 | Ravi | 105 |
| 3 | Jignesh | 101 |
| 4 | Nihir | 103 |
| 5 | Sohil | 104 |

To display the name of players who plays CaromBoard.

ii). To display details of those game which are having PrizeMoney more than 8000.

iii). To display the details of those games whose name starts from character 'B'.

iv). To display sum of Prize Money for each of the Number of participation groupings.
(as shown in column Number 2 or 4)

8. Consider the following tables EMPLOYEE and DESIG. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii)

**EMPLOYEE**

| W_ID | FIRSTNAME | LASTNAME | CITY |
|---|---|---|---|
| 102 | SAM | TONES | PARIS |
| 105 | SARAH | ACKERMAN | NEW YORK |
| 144 | MANILA | SENGUPTA | NEW DELHI |
| 210 | GEORGE | SMITH | HOWARD |
| 255 | MARY | JONES | HUSTON |
| 300 | ROBERT | SAMUEL | WASHINGTON |
| 335 | HENRY | WILLIAMS | BOSTON |
| 400 | RONNY | LEE | NEW YORK |
| 451 | PAT | THOMPSON | PARIS |

**DESIG**

| W_ID | SALARY | BENEFITS | DESIGNATION |
|---|---|---|---|
| 102 | 75000 | 15000 | MANAGER |
| 105 | 85000 | 25000 | DIRECTOR |
| 144 | 70000 | 15000 | MANAGER |
| 210 | 75000 | 12500 | MANAGER |
| 255 | 50000 | 12000 | CLERK |
| 300 | 45000 | 10000 | CLERK |
| 335 | 40000 | 10000 | CLERK |
| 400 | 32000 | 7500 | SALESMAN |
| 451 | 28000 | 7500 | SALESMAN |

**i).** Display FirstName and City of Employee having salary between 50,000 and 90,000

ii). Display details of Employees who are from "PARIS" city.

iii). Increase the benefits of employee having W_ID = 210 by 500.

iv). Count number of employees whose name starts from character 'S'.

v). Select MAX(salary) from desig;

vi). Select FirstName from employee, desig
where designation = 'MANAGER' AND employee.W_ID = desig.W_ID;

vii). Select COUNT (DISTINCT designation) from desig;

viii). Select designation, SUM(salary) from desig
Group by designationHaving count (*) > 2;

MS-23L

# Karnataka State Open University

M.Sc (IT) II Semester Practical List [New Scheme]

Time: 3hrs OOAD and UML Max Marks: 100

Write Programs using C++ /Java using OOAD concepts

1. Write a program to demonstrate multiple inheritances.
2. Write a program to demonstrate multi level inheritance.
3. Write a program to demonstrate Overriding.
4. Write a program to demonstrate overloading.
5. Write a program to demonstrate package.
6. Write a program to demonstrate message passing between two objects.
7. Write a program to demonstrate UML concepts like association and aggregation.
8. Write a program to demonstrate UML concepts like composition and dependency.
9. Write a program to demonstrate UML concepts like realization and relationship between two classes.
10. Write a program to demonstrate UML concepts like collaboration.
11. Write a program to demonstrate persistence.
12. Write a program to demonstrate generalization and dependency between two classes.
13. Write a program to demonstrate relations among two objects.
14. Write a program to demonstrate interface and realization.
15. Write a program to demonstrate association stereotypes.

MS-24L

# Karnataka State Open University

M.Sc (IT) II Semester Practical List[New Scheme]

Time : 3hrs Advanced Java Max Marks : 100

## Write Programs and Execute

1. Write a java exception handling program to demonstrate checked Exceptions.
2. Write a java exception handling program to demonstrate unchecked Exceptions.
3. Write a java program Thread life cycle also assign thread priority and display it.
4. Write a java applet to implement applet life cycle.
5. Write a java applet to implement demonstrate simple calculator.
6. Write java applet to demonstrate awt components which should have checkbox, radio button, textbox and labels.
7. Write a java networking program to demonstrate client server interaction.
8. Write a JDBC program to connect the database and verify the username and password from the database.
9. Write a java program to implement server interface using RMI.
10. Write a java program to implement insert and delete queries using JDBC.
11. Write a java Servlet program to demonstrate Hello Message .
12. Write a java servlet to demonstate data management using IO streams.
13. Write a java servlet program to demonstrate client interaction using doget() and dopost() methods.
14. Write a JSP program to demonstrate and to display forwarded information in a separate jsp subpage.
15. Write a java program to get the input from JSP page and to generate result to a java servlet.