**AMITY UNIVERSITY BENGALURU**

**KARNATAKA**

**PROGRAMMING IN C & C++**

**CSE1017**



|  |  |
| --- | --- |
| **SUBMITTED TO :**  Dr. Punam Kumari  Associate Professor & Program Lead  Computer Science & Engineering Dept. | **SUBMITTED BY:**  <Student Name>  <Enrollment Number>  <Batch No.> |

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Program Name** | **Date** | **Page No.** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Module 1: Introduction to C**

**1. Case: Temperature Alert System**

Write a C program that takes temperature input (integer) from the user and categorizes it as:

"Cold" if temperature < 15

"Normal" if 15 ≤ temperature ≤ 30

"Hot" if temperature > 30

Use conditional (ternary) operators and appropriate data types.

**Code:**

#include <stdio.h>

int main() {

int temperature;

printf("Enter the temperature: ");

scanf("%d", &temperature);

if (temperature < 15) {

printf("Cold\n");

} else if (temperature <= 30) {

printf("Normal\n");

} else {

printf("Hot\n");

}

return 0;

}

**Output:**

Enter the temperature: 35

Hot

**2. Case: Employee Tax Calculation**

Write a C program to calculate the net salary of an employee. Use constants for:

Base salary = 20000

HRA = 5000

Tax rate = 10%

Use appropriate constants (#define or const) and arithmetic expressions. Display gross and net salary.

**Code:**

#include <stdio.h>

int main() {

int base\_salary = 20000;

int hra = 5000;

float tax\_rate = 0.10;

float gross\_salary, tax, net\_salary;

gross\_salary = base\_salary + hra;

tax = gross\_salary \* tax\_rate;

net\_salary = gross\_salary - tax;

printf("Gross Salary = %.2f\n", gross\_salary);

printf("Net Salary = %.2f\n", net\_salary);

return 0;

}

**Output:**

Gross Salary = 25000.00

Net Salary = 22500.00

**3. Case: Scope & Lifetime**

Write a C program that declares a global counter variable, a local variable inside main(), and a static variable inside a function. Call the function multiple times to observe how the static variable behaves. Print all variable values each time. Demonstrate understanding of scope and lifetime of variables.

**Code:**

#include <stdio.h>

int global\_counter = 1;

void myFunction() {

static int static\_var = 1;

printf("Global counter = %d\n", global\_counter);

printf("Static variable = %d\n", static\_var);

static\_var = static\_var + 1;

}

int main() {

int local\_var = 10;

printf("Local variable = %d\n", local\_var);

myFunction();

global\_counter = global\_counter + 1;

myFunction();

global\_counter = global\_counter + 1;

myFunction();

return 0;

}

**Output:**

Local variable = 10

Global counter = 1

Static variable = 1

Global counter = 2

Static variable = 2

Global counter = 3

Static variable = 3

**4. Case: Result Evaluation**

Write a C program to accept marks of 3 subjects. Calculate average using an arithmetic expression. Use relational and logical operators to determine if:

Student has passed (each subject ≥ 40)

Grade is Distinction (average ≥ 75)

Display an appropriate message.

**Code:**

#include <stdio.h>

int main() {

int mark1, mark2, mark3;

float average;

printf("Enter marks for 3 subjects: ");

scanf("%d %d %d", &mark1, &mark2, &mark3);

average = (mark1 + mark2 + mark3) / 3.0;

if (mark1 >= 40 && mark2 >= 40 && mark3 >= 40) {

printf("Result: Pass\n");

if (average >= 75) {

printf("Grade: Distinction\n");

}

} else {

printf("Result: Fail\n");

}

return 0;

}

**Output:**

Enter marks for 3 subjects: 35 70 80

Result: Fail

**Module 2: Conditional Statements & Loops**

**1. Pattern Printing Using Nested For Loops**

Write a program to print the following pattern (for n rows):

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\*

**Code:**

#include <stdio.h>

int main() {

int i, j, n;

printf("Enter number of rows: ");

scanf("%d", &n);

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

for (j = 1; j <= n; j++) {

if (i == 1 || i == n || j == 1 || j == n) {

printf("\*");

} else {

printf(" ");

}

}

printf("\n");

}

return 0;

}

**Output:**

Enter number of rows: 5

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\*

**2. Matrix multiplication Using Two-Dimensional Arrays and Nested Loops**

Write a program to multiply two matrices using two-dimensional arrays and nested for loops.

**Code:**

#include <stdio.h>

int main() {

int a[10][10], b[10][10], c[10][10];

int r1, c1, r2, c2;

int i, j, k;

printf("Enter rows and columns of first matrix: ");

scanf("%d %d", &r1, &c1);

printf("Enter rows and columns of second matrix: ");

scanf("%d %d", &r2, &c2);

if (c1 != r2) {

printf("Multiplication not possible.\n");

return 0;

}

printf("Enter elements of first matrix:\n");

for (i = 0; i < r1; i++) {

for (j = 0; j < c1; j++) {

scanf("%d", &a[i][j]);

}

}

printf("Enter elements of second matrix:\n");

for (i = 0; i < r2; i++) {

for (j = 0; j < c2; j++) {

scanf("%d", &b[i][j]);

}

}

for (i = 0; i < r1; i++) {

for (j = 0; j < c2; j++) {

c[i][j] = 0;

for (k = 0; k < c1; k++) {

c[i][j] = c[i][j] + a[i][k] \* b[k][j];

}

}

}

printf("Result matrix is:\n");

for (i = 0; i < r1; i++) {

for (j = 0; j < c2; j++) {

printf("%d ", c[i][j]);

}

printf("\n");

}

return 0;

}

**Output:**

Enter rows and columns of first matrix: 2 3

Enter rows and columns of second matrix: 3 2

Enter elements of first matrix:

1 2 3

4 5 6

Enter elements of second matrix:

7 8

9 10

11 12

Matrix A (2x3):

1 2 3

4 5 6

Matrix B (3x2):

7 8

9 10

11 12