**Exercise**

**Theory Questions**

1. Write four advantages of using functions.

**Advantages of using functions:**

* Code reusability
* Easier debugging and maintenance
* Modular programming structure
* Reduces redundancy

1. Explain why some functions do not have a **return**statement.

· Functions with void return type do not return a value.

· Some functions perform tasks without needing to return data (e.g., printf() or exit()).

1. Explain the different between a local variable and global variable.

· Local variable: Declared inside a function, accessible only within that function.

· Global variable: Declared outside all functions, accessible throughout the program.

1. Describe the recursion?

· Recursion is when a function calls itself to solve a problem by breaking it into smaller instances.

· Example: Factorial calculation using recursion.

1. What is difference between the users-define and pre-define functions in C/C++.

**User-defined vs. Predefined functions:**

* **User-defined functions**: Created by the programmer for specific tasks (e.g., int add(int a, int b)).
* **Predefined functions**: Built-in functions from standard libraries (e.g., printf(), sqrt()).

**Practical Questions**

1. Write a program to display your complete profile, using at least 3 functions.

**Hint:** If user press 1: your bio-data will display onto the screen

If user press 2: your academic profile will display onto screen

#include <stdio.h>

using namespace std;

void bioData() {

cout << "Name: John Doe\nAge: 20\n";

}

void academicProfile() {

cout << "Degree: Computer Science\nCollege: XYZ University\n";

}

int main() {

int choice;

cout << "Press 1 for Bio-data, 2 for Academic Profile: ";

cin >> choice;

if (choice == 1) bioData();

else if (choice == 2) academicProfile();

else cout << "Invalid choice";

return 0;

}

1. Write a program to find;
2. Surface area (A=4pi r 2) **b)** Area of rectangle (A= width x Length).

#include <iostream>

#include <cmath>

using namespace std;

double surfaceArea(double r) {

return 4 \* M\_PI \* r \* r;

}

double rectangleArea(double width, double length) {

return width \* length;

}

int main() {

cout << "Surface Area: " << surfaceArea(5) << endl;

cout << "Rectangle Area: " << rectangleArea(4, 6) << endl;

return 0;

}

1. Write a function to calculate if a number is prime, Return 1 if it is prime and 0 if it is not a prime.

bool isPrime(int n) {

if (n < 2) return false;

for (int i = 2; i \* i <= n; i++)

if (n % i == 0) return false;

return true;

}

1. Write a program to calculate factorial of a number input by the user using by Recursion.

int factorial(int n) {

return (n == 0) ? 1 : n \* factorial(n - 1);

}

1. Make a calculator using functions **sum( ),** **sub( ), mul( ), div( )** for calculations.

int sum(int a, int b) { return a + b; }

int sub(int a, int b) { return a - b; }

int mul(int a, int b) { return a \* b; }

float div(float a, float b) { return (b != 0) ? a / b : 0; }

1. Write a function **pow**(x,y) to calculate the value of x raised to y.

int sum(int a, int b) { return a + b; }

int sub(int a, int b) { return a - b; }

int mul(int a, int b) { return a \* b; }

float div(float a, float b) { return (b != 0) ? a / b : 0; }

**Objective MCQ’s**

1. A variable that is declared outside a function is called a \_\_\_\_ variable.
2. Local
3. Global
4. Program
5. Class
6. A local variable must be declared \_\_\_\_\_\_\_\_\_\_.
7. Before a function
8. After a function
9. Within the braces of a function definition.
10. With the local keyword.
11. A (n) \_\_\_\_\_\_ allows you to treat a related group of C/C++ commands as a single unit.
12. Statement
13. Variable
14. Event
15. Function

4. When create user-define function should be use \_\_\_\_\_\_\_ with function name.

1. function
2. function return type and function parameters (if any)
3. procedure
4. nothing all

5. The function calls itself this process is called\_\_\_\_\_\_\_\_.

1. Repetitive
2. Decision
3. Recursion
4. loop
5. A program module in C/C++ is called \_\_\_\_\_\_\_\_\_\_.
6. Variable
7. Function
8. Sub-routine
9. Event
10. Function is invoked with a \_\_\_\_\_\_\_\_\_.
11. Return to function.
12. Function Definition.
13. Function Call.
14. Function Prototype.
15. A global variable is defined in a declaration.
16. In main () only
17. In the first function that uses it
18. Outside the any function
19. Outside the main function.
20. A variable that is known only with in the function in which it is defined is called a\_\_\_\_\_\_.
21. Global variable
22. Function
23. Local variable
24. Variable scope
25. Which of these are valid reasons **f**or using functions?
26. They use less memory than repeating the same code.
27. They run faster.
28. They keep different program activities separate.
29. They keep variables safe from other parts of the program.