Instruction Hours / week:L: 0 T: 0 P: 4 Marks: Internal: 40 External: 60 Total: 100 End Semester Exam: 3 Hours

Course Objectives

- To develop programming skills using the fundamentals and basics of C Language.
- To enable effective usage of arrays, structures, functions, pointers and to implement the memory management concepts.
- To teach the issues in file organization and the usage of file systems.
- To learn the characteristics of an object-oriented programming language: data abstraction and information hiding, inheritance, and dynamic binding of the messages to the methods.

Course Outcomes (COs)

After the completion of this course, a successful student will be able to do the following:

- 1. Develop programs using the basic elements like control statements, Arrays and Strings .
- 2. Solve the memory access problems by using pointers
- 3. Learn the basics of file handling mechanism that is essential for understanding the concepts in database management systems.
- 4. Understand the uses of preprocessors and various header file directives.
- 5. Use the characteristics of an object-oriented programming language in a program.

List of Programs

- 1. WAP to print the sum and product of digits of an integer.
- 2. WAP to reverse a number.

3. Write a function that checks whether a given string is Palindrome or not. Use this function to

find whether the string entered by user is Palindrome or not.

4. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.

5. WAP to print a triangle of stars as follows (take number of lines from user):

* *** ***** ******* 6. Write a program that swaps two numbers using pointers.

7. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.

8. Write a menu driven program to perform following operations on strings:

- a) Show address of each character in string
- b) Concatenate two strings without using streat function.
- c) Concatenate two strings using streat function.
- d) Compare two strings
- e) Calculate length of the string (use pointers)
- f) Convert all lowercase characters to uppercase
- g) Convert all uppercase characters to lowercase
- h) Calculate number of vowels
- i) Reverse the string

9. WAP to display Fibonacci series (i)using recursion, (ii) using iteration

10. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration

11. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).

12. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

13. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.

14. Copy the contents of one text file to another file, after removing all whitespaces.

15. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.

1.Write a program to find sum and product of digits.

Aim:

To write a program to find sum and product of digits.

ALGORITHM

- STEP 1: Start the program.
- STEP 2: To get the input for a number with more than 1 digit.
- STEP 3: Get the input for a number using int variable name as n. And also assign sum=0; and product=1;
- STEP 4: Extract the digits of the number using modulo division is n%10.
- STEP 5: Find the product of the digits by using formula applied in while loop calculate product=product*(n%10)
- STEP 6: Find the sum of digits by using condition sum=sum+n%10
- STEP 7: Print the product of digit and sum of digit using cout<< sum;
- STEP 8: Stop the program.

```
Program coding:
#include<iostream.h>
#include<conio.h>
void main()
{
int n;
int sum=0,product=1;
clrscr();
cout<<"\n To calculate sum and product of digits";</pre>
cout<<"\n Enter the digits:";
cin>>n;
while(n!=0)
ł
sum=sum+n%10;
product=product*(n%10);
n=n/10;
}
cout<<"\n Sum of all digit is "<<sum;
cout<<"\n Product of all digit is"<<pre>product;
getch();
}
```

calculate sum and product of digits

Enter the digits :

5

6

Sum of all digit is =11

Product of all digit is =30

Result:

Ex . No: 2REVERSE OF THE GIVEN NUMBERDATE:

Aim:

To write a C++ program to print reverse of the given number.

STEP 1:Start the program.

STEP 2 : Get the input for a number 'n' Value declared as integer.

STEP 3 : Start a loop and continue until the number becomes 0.

STEP 4 : Extract the digits using integer division and calculate the reverse

value using formula r=n%10 and n=n/10; of the number stored in 'r'

STEP 5 : Print the reversed number using cout<< r; statement.

STEP 6 : Stop the program.

```
#include<iostream.h>
#include<conio.h>
void main()
{
int k;
int rev=0;
clrscr();
cout<<"\n Reverse of the number";
cout<<"\n enter the digits:";</pre>
cin>>k;
while(k!=0)
{
rev=rev*10;
rev=rev+k%10;
k=k/10;
}
cout<<"Reversed number is:"<<rev;
getch();
}
```

Reverse a number

Enter the digits:98

Reversed number:89

Result:

PALINDROME USING FUNCTIONS

AIM

To write a program that checks whether a given string is palindrome or not.

ALGORITHM

STEP 1 : Start the program.

STEP 2 : Declare a character array word.

STEP 3 : Get the input string in the word.

STEP 4 :Pass the entered string to the function check()

STEP 5:Inside the function check, declare and calculate len as len = strlen(word)-(index+1)

STEP 6 :using an If statement compare the first and last character of the string.

STEP 7 :Ifall the characters match in reverse then print the result as " the entered string is a Palindrome".

STEP 8 : If the string a is not equal to string b then print the result as "Not a Palindrome".

STEP 9 : Stop the program.

PALINDROME USING FUNCTIONS

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void check(char[],int);
void main()
{
char word[15];
clrscr();
cout<<"\n\n PALINDROME USING FUNCTION:";</pre>
cout<<"\n\n Enter a word to check if it is a palindrome\n";
cin>>word:
check(word,0);
getch();
void check(char word[],int index)
intlen=strlen(word)-(index+1);
if(word[index]==word[len])
if(index+1==len||index==len)
cout << "\n The entered word is a palindrome\n";
return;
}
check(word,index+1);
}
else
cout << "\n The entered word is not a palindrome\n";
```

}

OUTPUT:

Enter a word to check if it is a palindrome : Malayalam

The entered word is a palindrome

Enter a word to check if it is a palindrome : welcome

The entered word is not a palindrome

Result:

4.write a program to find the given number is prime or not.

Aim:

To write a program to find the given number is prime or not.

Algorithms:

Step 1: Start the program.

- Step 2: Declare variables n,sum=0,product=1.
- Step 3: Read values for sum and product.
- Step 4: Enter the two values.
- Step 5: Display sum and product of the value.
- Step 6: Stop the process.

#include <iostream>
using namespace std;

```
int main()
{
 int n, i;
 bool isPrime = true;
 cout << "Enter a positive integer: ";</pre>
 cin >> n;
 for(i = 2; i \le n / 2; ++i)
 {
   if(n \% i == 0)
    {
      isPrime = false;
      break;
   }
 }
 if (isPrime)
   cout << "This is a prime number";</pre>
 else
   cout << "This is not a prime number";</pre>
 return 0;
```

}

Enter a positive integer: 23

This is a prime number

Result:

.

5.Write a program to print triangle of stars.

Aim:

To write a program to print triangle of stars.

Algorithm:

Step 1: Start the program.

Step 2: Declare variables rows, I, j, space.

Step 3: Read values for rows and space.

Step 4: Enter the values to print.

Step 5: Display the triangle of stars.

Step 6: Stop the process.

```
#include <iostream>
using namespace std;
int main()
{
int rows, i, j, space;
cout << "Enter number of rows: ";</pre>
cin >> rows;
for(i = 1; i <= rows; i++)
{
for(space = i; space < rows; space++)</pre>
{
cout << " ";
}
for(j = 1; j <= (2 * i - 1); j++)
{
cout << "*";
}
cout \ll "\n";
}
return 0;
}
```

nter number of rows: 6

*

Result:

6.Write a program to swap the numbers.

Aim:

To write a program to swap the numbers.

Algorithm:

Step 1: Start the program.

Step 2: Declare variables a,b,temp.

Step 3: Read values for a and b.

Step 4: Enter the values for swapping.

Step 5: Display after swapping of the value.

Step 6: Stop the process.

#include <stdio.h>

```
#define SWAP(x, y) (x ^{=} y ^{=} x ^{=} y)
```

```
int main()
```

```
{
```

int num1, num2;

printf("Enter any two number to swap: ");

scanf("%d%d", &num1, &num2);

printf("Values before swapping\n");

printf("num1 = %d, num2 = %d\n\n", num1, num2);

SWAP(num1, num2);

```
printf("Values after swapping\n");
```

printf("num1 = %d, num2 = %d\n", num1, num2);

return 0;

}

Enter any two number to swap: 10 20

Values before swapping

num1 = 10, num2 = 20

Values after swapping

num1 = 20, num2 = 10

Result:

7.Write a program to find the radius of a circle.

Aim:

To write a program to find the radius of a circle.

Algorithm:

- Step 1: Start the program.
- Step 2: Declare variables radius.
- Step 3: Read values for find out radius of circle.
- Step 4: Enter the values.
- Step 5: Display radius of circle.
- Step 6: Stop the process.

#include<stdio.h>

int main()

{

int radius;

float PI=3.14, area, ci;

printf(" Enter the Radius of Circle: ");

scanf("%d",&radius);

area = PI * radius * radius;

printf("\n Area of a Circle : %f ",area);

ci = 2 * PI * radius;

printf("\n Circumference of a Circle is : %f ",ci);

return(0);

}

Enter the Radius of Circle: 3

Area of a Circle : 28.260000

Circumference of a Circle is : 18.840000

Result:

8.write a program to perform string operations.

Aim:

To write a program to perform string operations.

Algorithms:

Step 1: Start the program.

Step 2: Declare int main and string header file.

Step 3: Read values using string functions to print.

Step 4: Enter the two values.

Step 5: Display string of the values.

Step 6: Stop the process.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main()
{
clrscr();
char s1[80], s2[80];
strcpy(s1,"c++");
strcpy(s2,"is power programming");
cout<<"length:"<<strlen(s1);</pre>
cout<<' ' << strlen(s2) << '\n';
if(!strcmp(s1,s2))
cout << "The string are equal\n";
else
cout << "not equal\n";</pre>
strcat(s1,s2);
cout \ll s1 \ll n';
strcpy(s2,s1);
cout<<s1<<" and "<<s2<< "\n";
if(!strcmp(s1,s2))
cout<<"s1 and s2 are now the same.\n";
getch();
}
```

Length: 3 20

not equal

C++ is power programming

C++ is power programming and C++ is power programming

s1 and s2 are now the same.

Result:

9.write a program to find Fibonacci series using recursion

Aim:

To write a program to find Fibonacci series using recursion iteration.

Algorithm:

Step 1: Start the program.

Step 2: Declare variables n1=0,n2=1,n3,i,number.

Step 3: Read values that need to be executed.

Step 4: Enter values.

Step 5: Display Fibonacci series.

Step 6: Stop the process.

}

```
#include<iostream.h>
#include<conio.h>
int fibonacci(int n)
{
if((n==1)||(n==0))
ł
return(n);
}
else
{
return(fibonacci(n-1)+fibonacci(n-2));
}
}
void main()
{
int n,i=0;
clrscr();
cout << "\nFIBBONACCI SERIES USING RECURSION";
cout << "\n ******* ****** *****************;
cout<<"\n\n input the number of terms for fibonacci series:";
cin>>n;
while(i<n)
{
cout<<"\n"<<fibonacci(i);
i++;
}
getch();
```

FIBBONACCI SERIES USING RECURSION

Input the number of terms for fibonacci series: 7

0			
1			
1			
2			
3			
5			

8

Result:

10.Write a program to find factorial using recursion and iteration.

Aim:

To write a program to find factorial using recursion and iteration.

Algorithm:

Step 1: Start the program.

Step 2: Declare variables n.

Step 3: Read values using positive integer.

Step 4: Enter values.

- Step 5: Display factorial using recursion.
- Step 6: Stop the process.

```
#include<iostream.h>
#include<conio.h>
int factorial(int n);
void main()
{
int n;
clrscr();
cout<<"\n\n FACTORIAL USING RECURSION \n";
cout<<"\n Enter a positive integer:";
cin>>n;
cout<<"\n\n factorial of"<< n <<"="<<factorial(n);</pre>
getch();
}
int factorial(int n)
{
if(n!=1)
n=n*factorial(n-1);
return n;
}
```

FACTORIAL USING RECURSION

Enter a positive integer : 5

factorial of 5 : 120

Result:

11.Write a program to perform simple inheritance.

Aim:

To write a program to perform simple inheritance.

Algorithms:

Step 1: Start the program.

Step 2: Declare using namespace std.

Step 3: Read values that need to be executed.

Step 4: Enter input values.

Step 5: Display simple inheritance.

Step 6: Stop the process.

```
#include <iostream.h>
using namespace std;
class Parent
{
  public:
   int id_p;
};
class Child : public Parent
{
  public:
   int id_c;
};
int main()
  {
    Child obj1;
    obj1.id_c = 7;
    obj1.id p = 91;
    cout << "Child id is " << obj1.id_c << endl;
    cout << "Parent id is " << obj1.id_p << endl;
    return 0;
  }
```

Child id is 7 Parent id is 91

Result:

12.Write a program to perform method overloading.

Aim:

To write a program to perform method overloading.

Algorithm:

Step 1: Start the program.

- Step 2: Declare datatypes int and float
- Step 3: Read values that need to be executed.
- Step 4: Enter input values.
- Step 5: Display method overloading.
- Step 6: Stop the process.

#include <iostream>
using namespace std;

int absolute(int);
float absolute(float);

```
int main() {
  int a = -5;
  float b = 5.5;
  cout << "Absolute value of " << a << " = " << absolute(a) << endl;
  cout << "Absolute value of " << b << " = " << absolute(b);
  return 0;
}
int absolute(int var) {
  if (var < 0)
     var = -var;
  return var;
}
float absolute(float var){
  if (var < 0.0)
     var = -var;
  return var;
}
```

Absolute value of -5 = 5

Absolute value of 5.5 = 5.5

Result:

13.Write a program to collect student details using structure.

Aim:

To .write a program to collect student details using structure

Algorithms:

Step 1: Start the program.

- Step 2: Declare struct to collect student details.
- Step 3: Read values that need to be executed.
- Step 4: Enter input values.
- Step 5: Display the student details.
- Step 6: Stop the process.

```
#include<stdio.h>
struct student
{
char name[50];
int roll;
int marks;
};
void main()
ł
struct students s[10];
int I;
printf("enter information of students:\n");
for(i=0;i<3;++i)
{
s[i].roll=i+1;
printf("\nfor roll number%d\n",s[i].roll);
printf("enter name:");
scanf("%s",s[i].name);
printf("enter marks:");
scanf("%d",&s[i].marks);
printf(\n");
printf("displaying information of students:\n\n");
for(i=0;i<=3;++i)
{
printf("\n information for roll number%d:\n",i+1);
printf("name:");
puts(s[i].name);
printf("marks:%d",s[i].marks);
}
getch();
```

For roll number 1 enter name:hari enter mark :56 For roll number 2 Enter name:surya enter mark:89 for roll number3 enter name:bharathi enter marks:89 Displaying information of students: information for roll number 1: name: hari mark:56 information for roll number 2: name: surya marks:89 information for roll number 3: name: bharathi marks:89

Result:

14.Write a program to copy the contents of a text file.

Aim:

To write a program to copy th contents of a text file.

Algorithm:

Step 1: Start the program.

Step 2: Declare the FILE *fptr1, *fptr2.

Step 3: Read values that need to be executed.

Step 4: Enter input values.

Step 5: Display simple inheritance.

Step 6: Stop the process.

```
#include<stdio.h>
#include<conio.h>
main(int arg c,char *arg[])
{
FILE *fp1-*fp2;char ch;
if (argc!=3)
{
printf("insufficient arguments");
exit(0);
}
fp1=fopen(argv[1],"r");
fp2=fopen(argv[2],"w");
if(fp1=NULL||fp2==NULL)
{
printf("unable to open file");
exit(0);
}
while(!feof (fp1));
h1=fgetc(fp1);
fput((ch),fp2);
printf("file copied successfully");
fclose(fp1);
fclose(fp2);
getch();
```

C:\temp:> cmd f1.txt f2.txt

File copied successfully

Result:

15.Write a program to reverse the element of an array.

Aim:

To write a program to reverse the element of an array.

Algorithm:

Step 1: Start the program.

Step 2: Declare the int n, c, d, a[100], b[100];

Step 3: Read values that need to be executed.

Step 4: Enter input values.

Step 5: Display reverse array using for (c = 0; c < n; c++).

Step 6: Stop the process.

```
#include <stdio.h>
int main()
{
 int n, c, d, a[100], b[100];
 printf("Enter the number of elements in array\n");
 scanf("%d", &n);
 printf("Enter array elements\n");
  for (c = 0; c < n ; c++)
   scanf("%d", &a[c]);
  for (c = n - 1, d = 0; c >= 0; c--, d++)
   b[d] = a[c];
 for (c = 0; c < n; c++)
   a[c] = b[c];
  printf("Reverse array is\n");
  for (c = 0; c < n; c++)
   printf("%d\n", a[c]);
  return 0;
```

}

Enter the number of elements in array

Enter the array elements

Result: