Brainware Computer Academy

Programming With C

(Specially Designed for WBCHSE, CBSE, ICSE & ISC)

(Practical Assignment)

Session 2:

1. Draw a flow chart which will calculate the following sum of series

   a. S = 1 + 2 + 3 + ... + 10
   b. S = 1 + 3 + 5 + ... + 10
   c. S = 0 + 1 + 1 + 2 + 3 + ... upto a specified term.

2. Type the following codes and watch the output (compile and run).

```
#include <stdio.h>
int main()
{
    int a=10;
    char b=65, c=’x’;
    printf(“%d %c %c\n”, a, b, c);
}
```

Session 3:

3. Make the following changes in the above program and watch the output:

   a. Add the following lines at the end

```
c = a + b;
printf(“%c\n”, c);
printf(“%d\n”, c);
```

   b. Declare a new variable “sum1” as integer at the beginning of main() and add the following lines at the end

```
sum1=a+25;
printf(“%d\n”, sum1);
```

   c. Declare a variable “sum2” similarly and add the following lines at the end

```
int  sum2=a+25.66;
printf(“%f\n”,sum2);
printf(“%d\n”,sum2);
printf(“%c\n”,sum2);
```

   d. Modify the “printf” statement in the following way

```
printf(“\n%f”,sum2++);
```

   e. Do the same by adding “\t”.

```
printf(“\t%f”,++sum2);
```

4. Type the following program and watch the output

```
#include<stdio.h>
int main()
{
      int a, b;
      printf(“enter two nos: “);
```

```
        scanf(“%d %d”, &a, &b);
        printf(“\nResult = %d”, a*b);
        printf(“\nResult = %d”, a+b);
        printf(“\nResult = %d”, a/b);
}
```

5. Accept values into integer variables x and y from the console and display the result of the following equation
   $(X+Y)^2 – (X-Y)^2 – 4XY = ?$

6. Convert Fahrenheit degrees into Centigrade (taking input from keyboard formula: C/5 = (F-32)/9.

7. Take two numbers into two integer variables x and y and interchange their contents.

Session 4:

8. Watch the output of the following codes:
   a.
```
int main()
{
        int a=5;
        if (a>5)
        {
                puts(“good”);
                puts(“boy”);
        }
        else
        {
                 puts(“bad boy”);
        }
}
```

   b.
```
int main()
{
         int j=65;
         if (j>=65)
         {
                 if (j<=65)
                 {
                        printf(“good”);
                        printf(“girl”);
                 }
          }
}
```

9. Write a program that will accept a year (4 digit number) and will check whether it is a leap year or not.

10. Write a program that will take age and will display the following according to the input
    between 1 – 10 ➔ Child
    between 11 – 50 ➔ Adult
    over 50 ➔ Old

11. Take a number from user and check whether it is odd or even.

12. Print a lowercase character (input from keyboard) into upper case and vice-versa.

Session 5:

13. Accept the employee Name, Basic Pay, and calculate the Gross Salary, where Gross Salary = Basic Pay + DA.
    For DA calculation consider the following conditions:
    If BasicPay is more than 5000, then DA is 20% of BasicPay.
    If BasicPay is more than 3000 but less than or equal to 5000, then DA is 15% of BasicPay.
    If BasicPay is less than or equal to 3000, then DA is 10% of BasicPay.

14. Write a program to calculate the summation of odd numbers upto 10 starting from number 1

15. Write a program which will calculate the summation of even numbers upto 10 starting from number 2

16. Accept a number and find out its factorial.

Session 6:

17. Take two numbers m and n and find out m^n

18. Write a menu driven calculator that offers user the following options:
    1........... ADD
    2........... SUBTRACT
    3............ MULTIPLY
    4............ DIVISION
    5............ EXIT.
    Accept two numbers and menu option from user and perform accordingly.

19. Take a number from user and find out its Square Root.

20. Accept a number n and find out the prime numbers between 1 to n.

Session 7:

21. Write a program that will calculate the sum of numbers divisible by five from 1 to 100.

22. Print the characters and their corresponding ASCII values from 0 to 255.

23. Print the following output on the screen.

```
a) *                     b) A
   *  *                     B   B
   *  *  *                  C   C   C
   *  *  *  *               D   D   D   D
```

Session 8:

24. Print the following output on the screen.

```
          1
       1  2
    1  2  3
 1  2  3  4
```

25. Print the following output on the screen.

```
        *
      *     *
  *     *      *
*    *      *     *
```

26. Print the following output on the screen.

```

         *
      *     *
   *          *
*                *
   *           *
      *     *
         *
```

27. Print the following output on the screen.

```
         A
      B  A  B
   C  B  A  B  C
D  C  B  A  B  C  D
```

Session 9:

28. Write a program, which will accept 5 random numbers into an Array of 5 elements (integer). Read and display the Array elements.

29. Write a program (using Array) to:
    a. Find out the max. of 10 nos. given by user.
    b. Find out the min. of 10 nos. given by user.

30. Write the above program to display the Array elements into sorted order.

Session 10:

31. Take two 3 X 3 matrices, obtain their additio n in a third matrix and print it.

32. Accept a word from the user and reverse it using array and print it.

33. Write a program to determine whether a word is palindrome (MADAM) or not.

Session 11:

34. Write a program to count the no. of vowels in a string.

35. Write a program that will take a string and will print it in the following manner. [Let input = AMIT]

```
A
A  M
A  M  I
A  M  I  T
```

Session 12:

36. Write a program, which will accept 5 numbers into an integer Array. Display the Array elements with their corresponding memory address side-by-side. Analyze the displayed addresses.

37. Write a program (using integer pointer) to find out the largest of n nos. given by user.

Session 13:

38. Write a program that will remove capital letters from a string. The String will be implemented by a char* where memory allocation for the pointer will be done dynamically.
39. Write a program that will store and display few names using 2D pointer.

Session 14:

40. Re-write the program no.18 using individual functions for each menu options.

41. Swap the content of two variables using a user-defined function (having arguments as call-by-value method) and see swapping effect in main function keeping a display function for the two variables.

Session 15:

42. Re-write the program no.41 using a user-defined function (having arguments as call-by-reference method) and see swapping effect in main function.
43. Write an accept_age(int*) function that will only accept a positive non-zero number or otherwise displays an error massage that age can't be negative.

Session 16:

44. Write a sort function to sort an integer array in descending order.

45. Write a function that will convert a string into uppercase.

46. Accept a sentence and print it with each word in the sentence in reverse order. (i.e. "LET US SEE THE POWER OF C" will be as "TEL SU EES EHT REWOP FO C").

47. Write a program that will take 6 names and will sort them in ascending order using a function.

Session 17:

48. Find out the length of a word using a user defined recursive function.

49. Write a function that will compress multiple spaces in a string to one.

50. Write a function: void copy(char *s, char *d); that copies one string into another.

51. Write a function to search the number of occurrences of a word in a sentence.

52. Write a function to concatenate two strings and return the resultant string.

Session 18:

53. Write a program using a structure called Employee which takes data from users and display the data:
    Employee structure contains
    employee_id integer type
    employee_name char* type
    basic_pay integer type
    employee_salary integer type
    dept_no integer type

54. Modify the above program to add some validations on.
    a) employee_id should be in the range 100 – 500.
    b) employee_name should not be left blank.
    c) employee_salary should be positive numbers only.

Session 19:

55. Build a program to take 10 records in an array from the above said Employee structure and to display all records.

56. Modify the above program and add a function sort_employee() that will sort (on employee_id) all records in the array of employee structure.

57. Modify the above program and add a function display_employee( ) that will show all the records in the following pattern for e.g.:

```
SL. NO          ID          NAME            SALARY
______________________________________________________
1               101         RAHUL           5551
```

Session 20:

58. Using the same structure defined above build a menu driven program to
    1………………….Add
    2………………….Show
    3………………….Delete
    4………………….Exit

one by one record in a single linked list.

Session 21:

59. Write a union having the following members:
    a. int
    b. char

    Now first enter values in those members and then display them. Justify the output.

60. Write an enum to represent the weekdays and use it in a program.
61. Write a macro that will evaluate the greater of 2 numbers.

Session 22:

62. Using the same structure defined in program no.53 build a program named ‘EmpAdd.c’, which will store all record in a data file named ‘Employee.dat’.

63. Build a program named ‘EmpMod.c’ that will ask for the employee id, search it and if found, it will display a message “Record Found” otherwise it will display “Record not Found”.

64. Build a program ‘EmpRepo.c’ to display records of all employees (the record should be ordered by department number).

65. Build a deletion program ‘EmpDel.h’ that will ask for employee id, search it and if found will delete the corresponding record.

Session 23:

66. Write a program that will display the content of a text file where the file name has been supplied as command line argument.

67. Write a file copy program where the source & destination file names have been supplied as command line arguments.

Session 24:

68. Modify all the files of Session22 (EmpAdd.c, EmpRepo.c, EmpDel.c, EmpMod.c) as per the instructions given below:

    a. Change the main() of each file [Rename the main() as]

    EmpAdd.c → emp_add()
    EmpMod.c → emp_mod()
    EmpDel.c → emp_del()
    EmpRepo.c → emp_repo()

69. Define a new file named ‘MyProgram.c’

    Include this definition to the top just after #include <stdio.h>

```
#include “EmpAdd.c”
#include “EmpMod.c”
#include “EmpDel.c”
#include “EmpRepo.c”
```

Define the menu in the following manner:

---

1. Employee Addition
2. Employee Modification
3. Employee Deletion
4. Display Employee Records
5. Exit

Enter Your Choice [1/2/3]:

---

70. Write a program, which will sort the above ‘Employee.dat’ data file on the field ‘basic’ in descending order.