



Name :
Roll No. :
Invigilator's Signature :

CS/M. Tech (CSE)/SEM-1/CSEM-104/2011-12

2011

**ADVANCED PROGRAMMING AND
PROBLEM SOLVING**

Time Allotted : 3 Hours

Full Marks : 70

The figures in the margin indicate full marks.

*Candidates are required to give their answers in their own words
as far as practicable.*

Answer any *two* questions from Group-A and *three* questions
from Group-B

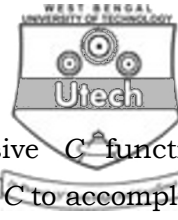
GROUP – A

Answer any *two* questions.

$2 \times 14 = 28$

1.
 - a) Define a class. How does it accomplish data hiding ?
 - b) When do we declare a member of a class static ? Give Example.
 - c) What is a parameterized constructor ? Explain with an example.
 - d) Can we have more than one constructors in a class ? If yes, explain the need for such a situation with suitable example.
 - e) What do you mean by friend function ? What is the significance of declaring a function as friend ? Give example.

$(2 + 2) + 2 + 2 + (1 + 2) + (1 + 2)$



2. a) Determine what the following recursive C function computes. Write an iterative function in C to accomplish the same purpose.

```
( int func ( int n )
{
    if ( n==0)
        Return (0);
    return (n + func (n-1));
}
```

- b) Write an iterative function in C to evaluate $a * b$ by using addition, where a & b are two non-negative integers.
- c) Define a recursive procedure. Write a recursive function GCD (x, y) in C that implements the division algorithm to compute GCD of two integers x & y . Rewrite this function in iterative form.

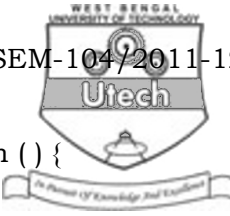
$$(2 + 3) + 3 + (2 + 2 + 2)$$

3. a) The 'C' in built function "strtok" works in the following manner :

```
/* usage of "strtok" */
char * first = strtok (mystring, ":");
while (first = strtok (NULL, ":"))
    print ("%s", first);
```

For an input to mystring = "Arindam : 40 : 500 : 5'7" " output will be respectively "Arindam" "40" "500" "5'77" "

Implement a function "my-strtok" which has similar semantics as the in built function "strtok".



```

b)  int * fun ( ) {
      int a ;
      a = 5;
      printf ("%d\n", a);
      return & a ;
    }

    void main ( ) {
      int * P ;
      P = fun ( );
      printf ("*P is = %d\n", *P);
    }

```

i) What will be the output to the above code ? Why ?

ii) The Code inside fun () is changed as follows :

```

int * fun ( ) {
  int * a ;
  *a = malloc ( sizeof (int));
  printf ("a is = %d"; *a);
  return a;
}

```

What will be that changed output & why ? 6 + (4 + 4)

```

4. a) class Parent {
      Public :
      Parent () {}
      ~Parent () { cout << "I am parent destructor" << endl; }
    };

    //=====

    class Child: public Parent {
      public :
      Child() {}
      ~Child() { cout<< "I am child destructor" << endl; }
    };

    //=====

```



```
void main () {  
    Parent *p [2];  
    Parent *pl = new Parent;  
    Child *cl = new Child;  
    p [0] = pl;  
    p[1] = cl;  
    for (int i=0 ; i < 2; i++)  
        delete p [i];  
}
```

- i) What will be the output to the above program ?
 - ii) What will be the changed output (if any) if the destructor in class Parent is declared as virtual ?
 - iii) How about Child destructor only declared as virtual (and not the Parent) ?
- b) What is wrong with the following code ?

```
Class Parent {  
    Public :  
    Parent () {...}  
    Parent (Parent obj) {......}  
};
```

Suggest suitable modifications to the above code to make it work correctly.

(3 + 3 + 3) + 5



GROUP – B

Answer any *three* questions.

3 × 14 = 42

5. a) The items : A, B, C, D and E are pushed in a stack, one after the other starting from A . The stack is popped four times and each element is inserted in a queue. Then two elements are deleted from the queue and pushed back on the stack. Now one item is popped from the stack. Which item will be popped and why ?
- b) A list of integers is read in, one at a time, and a binary search tree is constructed. Next the tree is traversed and the integers are printed. Which traversal would result in a print out which duplicates the original order of the list of integers ? Explain with an example.
- c) Write recursive routines in C to traverse a binary tree in Pre-order & Post-order.
- d) Define a Binary tree. Prove that a binary tree containing n nodes has exactly $(n + 1)$ NULL links.

$$2 + (1 + 3) + 5 + (1 + 2)$$



6. a) Eliminate recursion from the following code, applying standard procedures :

```
QSort (int lower, int upper) {  
    if (lower > upper) return;  
    int middle : = find Pivot (lower, upper);  
    QSort (lower, middle - 1);  
    QSort (middle + 1, upper);  
}
```

- b) Show with the following array that the non-recursive version indeed correctly implements the QuickSort procedure :

11 22 12 17 33 21 23 29 7 + 7

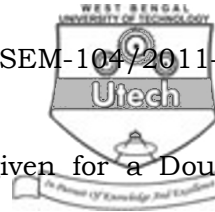
7. a) Define Binary Search Tree. When will a BST become an AVL tree ?

- b) Draw an AVL tree for the following key elements :

A, Z, B, Y, C, X, D, W, E, V, F

- c) Now delete a node from the above constructed tree which has both its children present.

- d) What is the purpose of a STACK in implementing a recursive procedure ? (2 + 2) + 6 + 2 + 2



8. a) The following Hash functions are given for a Double Hashing case where the hash table size is 13 :

$$h_1 = (\text{key} \bmod 13) \quad h_2 = 1 + (\text{key} \bmod 11)$$

Show the configuration of the Hash Table after the following values are inserted :

79 69 72 50 98

- b) Show how to implement a stack of integers in C by using an array S[STACKSIZE], where S[0] is used to contain the index of the top element of the stack & where S[1] thru S [STACKSIZE – 1] contain the elements of the stack.

7 + 7

=====