BITS PILANI, DUBAI CAMPUS
DUBAI INTERNATIONAL ACADEMIC CITY, DUBAI

FIRST SEMESTER 2023 – 2024
COURSE: CSF301 (Principles of Programming Languages)
COMPONENT: Tutorial Sheet 5 **DATE:** 19 October 2023

Q1. What is the output of following program? Assume address of x is 500 and integer is 4-byte size.

```
#include<stdio.h>
int main()
{
   int x=30, *y, *z;
   y=&x;    z=y;
   *y++=*z++;
   x++;
   printf("x=%d, y=%d, z=%d\n", x, y, z);
   return 0;
}
```

Ans: x=31, y=504, z=504

Q2. Add a statement in the function fun() such that address of a gets stored in j?.

```
#include<stdio.h>
int main()
{
   int *j;
   void fun(int**);
   fun(&j);
   return 0;
}
void fun(int **k)
{
   int a=10;
   /* Add a statement here */
}
```

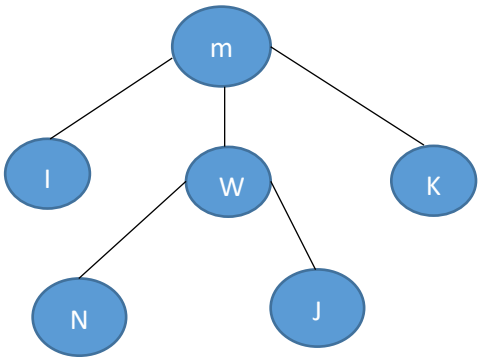
Ans: *k=&a

Q3. Draw the **Activation Tree** for the following skeletal C code **:**

```
W()
{
   N();
   J();
}
main()
{
```

```
  I(); W(); K();
}
```

Ans:



Q4. Find the output of following program.

```
/*  Finding OUTPUT OF C CODE: PARAMETER
PASSING  */
#include <stdio.h>
void main ()
{
  void fun(int xx, int *nn);
  int x[5], i;
  int n = 1;
  for (i = 0; i < 5; i += 1)
    {
      x[i] =  n + 2;
      fun(x[i], &n);
      n = n + 1;
    }
}

void   fun(int xx, int  *nn)
{
  int m, z;
  m = *nn + 5;
  z = xx + m;

  printf (" m = %d z= %d \n", m, z);
}
```

Ans:

m = 6 z= 9
m = 7 z= 11
m = 8 z= 13
m = 9 z= 15
m = 10 z= 17

Q5. Identify the parameter passing method and write the output of the program

```
void build_array( int array_variable[], int length_of_array )
    {
      for (int i=0; i<length_of_array; i++)
        {       array_variable[i] = i;
        }
```

```
        }
    int main()
        {
          int values[50];
          printf("the value at location 7 starts as %d\n", values[7]);
           build_array(values, 50);
           printf("the value at location 7 is now %d\n", values[7]);
          return 0;
        }
```

Ans: Call by reference

Output:

the value at location 7 starts as 0

the value at location 7 is now 7

Q6. Write the outcome of following code snippet by considering static scope and dynamic scope

```
void fun1(void);
void fun2(void);
int a = 1, b = 2, c = 3;
int main() {
   c = 4;
   fun1();
   return 0;
}
void fun1() {
   int a = 2, b = 3;
   fun2();
}
void fun2(){
   printf("%d %d %d", a, b, c);
}
```

Ans:

Output in Static scope: 1 2 4

Output in Dynamic scope: 2 3 4