

Name: Pranav Bhargava Enrollment No.: 9920103112
Batch: F4

DS Lab Assignment
(Week 1 – Lab A)

Memory Allocation – Static / Dynamic

Q1. Considering that an integer variable, a float variable, a double variable, a character variable, and a pointer variable need 4, 4, 8, 1, and 8 bytes memory space respectively, what will be the output of following C++ programs.

(a)	(b)
<pre>#include <iostream> using namespace std; class abc { int x; double y; }; int main() { abc *o1 = new abc, o2; cout<<"\nSize of o1 : "<<sizeof(o1); cout<<"\nSize of o2 : "<<sizeof(o2); cout<<"\nSize of abc is : "<<sizeof(abc); return 0; }</pre>	<pre>#include <iostream> using namespace std; class abc { int x; double y; int z; }; int main() { abc *o1 = new abc, o2; cout<<"\nSize of o1 : "<<sizeof(o1); cout<<"\nSize of o2 : "<<sizeof(o2); return 0; }</pre>
(c)	(d)

<pre> #include <iostream> using namespace std; class abc { int x; double y; </pre>	<pre> #include <iostream> using namespace std; class abc { float x; char y; </pre>
--	--

<pre> int z; int a; }; int main() { abc *o1 = new abc, o2; cout<<"\nSize of o1 : "<<sizeof(o1); cout<<"\nSize of o2 : "<<sizeof(o2); return 0; } (e) </pre>	<pre> int z; double a; }; int main() { abc *o1 = new abc, o2; cout<<"\nSize of o1 : "<<sizeof(o1); cout<<"\nSize of o2 : "<<sizeof(o2); return 0; } (f) </pre>
---	--

<pre> #include <iostream> using namespace std; class abc { char x[5]; double y; }; int main() { abc *o1 = new abc, o2; cout<<"\nSize of o1 : "<<sizeof(o1); cout<<"\nSize of o2 : "<<sizeof(o2); return 0; } </pre>	<pre> #include <iostream> using namespace std; class abc { char x[5]; float y[3]; }; int main() { abc *o1 = new abc, o2; cout<<"\nSize of o1 : "<<sizeof(o1); cout<<"\nSize of o2 : "<<sizeof(o2); return 0; } </pre>
---	---

Answer of 1)a:

```

C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe
Size of o1 : 8
Size of o2 : 16
Size of abc is : 16
Process returned 0 (0x0)   execution time : 0.074 s
Press any key to continue.
_

```

Answer of 1)b:

```

C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe
Size of o1 :8
Size of o2: 24
Process returned 0 (0x0)   execution time : 0.028 s
Press any key to continue.
_

```

Answer of 1)c:

```
C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe

Size of o1 :8
Size of o2: 24
Process returned 0 (0x0)    execution time : 0.029 s
Press any key to continue.
```

Answer of 1)d:

```
C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe

Size of o1 : 8
Size of o2 : 24
Process returned 0 (0x0)    execution time : 0.013 s
Press any key to continue.
-
```

Answer of 1)e:

```
C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe

Size of o1 :8
Size of o2: 16
Process returned 0 (0x0)    execution time : 0.028 s
Press any key to continue.
```

Answer of 1)f:

```
C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe

Size of o1 :8
Size of o2: 20
Process returned 0 (0x0)    execution time : 0.027 s
Press any key to continue.
-
```

Q2. Analyze the correctness and output of following programs

(a)	(b)
-----	-----

<pre> #include <iostream> #include <malloc.h> using namespace std; int main() { float *a; a = (float *)malloc(sizeof(int)); a[0] = 4.5; cout<<a[0]; return 0; } </pre>	<pre> #include <iostream> #include <malloc.h> using namespace std; int main() { int *a; a = (int *)malloc(sizeof(float)); a[0] = 5; cout<<a[0]; return 0; } </pre>
<p>(c)</p> <pre> #include <iostream> #include <malloc.h> using namespace std; int main() { int *a, *b; a = (int *)malloc(sizeof(int)); b = (int *)malloc(5*sizeof(int)); cout<<sizeof(a)<< sizeof(b); return 0; } </pre>	<p>(d)</p> <pre> #include <iostream> #include <malloc.h> using namespace std; int main() { int *a; a[0] = (int *)malloc(sizeof(int)); a[0] = 5; cout<<a[0]; return 0; } </pre>
<p>(e)</p> <pre> #include <iostream> #include <malloc.h> using namespace std; int main() { </pre>	<p>(f)</p> <pre> #include <iostream> #include <malloc.h> using namespace std; int main() { </pre>

<pre> int *a[5]; a[0] = (int *)malloc(sizeof(int)); a[0][0] = 5; cout<<a[0][0]; return 0; } </pre>	<pre> struct node{int a[10];}; struct node *n; n = (struct node *)malloc(sizeof(struct node)); cout<<sizeof(n); return 0; } </pre>
(g)	(h)
<pre> #include <iostream> #include <malloc.h> using namespace std; int main() { int *a[5]; a[0] = (int *)malloc(2*sizeof(int)); a[0][1] = 5; cout<<a[0][1]; return 0; } </pre>	<pre> #include <iostream> #include <malloc.h> using namespace std; int main() { int *a = (int *)malloc(5*sizeof(int)); a[0] = 1; a[1] = 2; a[2] = 3; a[3] = 4; a[4] = 5; delete(a); cout<<a[0]<<a[1]<<a[2]<<a[3]<<a[4]; return 0; } </pre>

Answer of 2)a:

```

C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe
4.5
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.

```

Answer of 2)b:

```
C:\Users\prana\Desktop\DSLlab\bin\Debug\DSLlab.exe
5
Process returned 0 (0x0)    execution time : 0.029 s
Press any key to continue.
_
```

Answer of 2)c:

```
C:\Users\prana\Desktop\DSLlab\bin\Debug\DSLlab.exe
88
Process returned 0 (0x0)    execution time : 0.029 s
Press any key to continue.
_
```

Answer of 2)d:

Incorrect statement, correct program:

```
#include <iostream>
#include <malloc.h>
using namespace std;
int main() {
int *a;
a =(int*)malloc(sizeof(int));
a[0] = 5;
cout<<a[0];
return 0;
}
```

Output:

```
C:\Users\prana\Desktop\DSLlab\bin\Debug\DSLlab.exe
5
Process returned 0 (0x0)    execution time : 0.015 s
Press any key to continue.
_
```

Answer of 2)e:

```
C:\Users\prana\Desktop\DSLlab\bin\Debug\DSLlab.exe
5
Process returned 0 (0x0)    execution time : 0.030 s
Press any key to continue.
_
```

Answer of 2)f:

```
C:\Users\prana\Desktop\DSLab\bin\Debug\DSLab.exe
8
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

Answer of 2)g:

```
C:\Users\prana\Desktop\DSLab\bin\Debug\DSLab.exe
5
Process returned 0 (0x0)   execution time : 0.032 s
Press any key to continue.
```

Answer of 2)h:

Incorrect statement, correct program:

```
#include <iostream>
#include <malloc.h>
using namespace std;
int main() {
int *a = (int *)malloc(5*sizeof(int));
a[0] = 1; a[1] = 2; a[2] = 3; a[3] = 4; a[4] = 5;
cout<<a[0]<<a[1]<<a[2]<<a[3]<<a[4];
delete (a);
return 0;
}
```

Output:

```
C:\Users\prana\Desktop\DSLab\bin\Debug\DSLab.exe
12345
Process returned 0 (0x0)   execution time : 0.027 s
Press any key to continue.
```

Q3. A dynamically created array stores following integer elements (odd and even integers)

2	8	3	6	7	9	5	4
---	---	---	---	---	---	---	---

It is desired to print/display the elements of this array in such manner that it first prints all the even elements then it prints all the odd elements. In above example, the displayed elements are as follows:

2 8 6 4 3 7 9 5

Write a program with and without STL to create the dynamic array of user inputted length (n), assign values at different indices of the array, and as presented in above example, display the elements of this array.

(Note: don't enter the elements manually, rather use following statement in loop to randomly assign elements (in range between 0 and 99) in the array: $A[i] = \text{rand()} \% 100$, where A is an array).

```
#include <iostream>
#include<stdlib.h>
#include<time.h>
using namespace std;
int main()
{
    srand(time(0));
    int n;
    cout<<"ENTER THE LENGTH OF THE ARRAY: ";
    cin>>n;
    int A[n];
    for(int x=0;x<n;x++)
        A[x]=0;
    int *Ar = (int *)malloc(n*sizeof(int)),c=0;
    int k=0;
    for(int i=0;i<n;i++)
    {
        int num=rand()%100;
        for(int j=0;j<n;j++)
        {
            if (num!=A[j])
                k++;
        }
        if(k==n)
        {
            A[i]=num;
            k=0;
        }
        else
        {
            i--;
            k=0;
        }
    }
}
```

```

    }
}
cout<<"THE ARRAY IS: ";
for(int j=0;j<n;j++)
    cout<<A[j]<<" ";
cout<<endl;
cout<<"ELEMENTS IN ORDER: ";
for(int i=0;i<n;i++)
{
    if(A[i]%2==0)
    {
        cout<<A[i]<<" ";
    }
    else
    {
        Ar[c]=A[i];
        c++;
    }
}
for(int j=0;j<c;j++)
    cout<<Ar[j]<<" ";
delete Ar;
}

```

Output:

```

C:\Users\prana\Desktop\DSLAb\bin\Debug\DSLAb.exe
ENTER THE LENGTH OF THE ARRAY: 8
THE ARRAY IS: 90 30 83 10 64 7 60 52
ELEMENTS IN ORDER: 90 30 10 64 60 52 83 7
Process returned 0 (0x0) execution time : 3.653 s
Press any key to continue.

```