**Course Name : Object Oriented Programming**

**Course Code : CSE 213**

**Instructor : MD. KHALID MAHMUD**

**MID- 01**

**Duration : 2 hours**

**Total Mark : 80**

**SPRING 2021**

**Q1: : [20 marks]**

Consider the following code. You need to write the COMPLETE C++ program:

```
class Truck {
private:
string engineType; string model, float price;
// define necessary methods, OR declare friends and define them globally,
// for input and output so that the given main() works.
};
//add necessary global functions (which are friends) so that the given main() works
int main(){
Truck *truckArr; int noOfTrucks, choice, i;
cout<<"How many trucks? "; cin>> noOfTrucks;
//add trucks to truckArr
for loop:{
addATruckToTruckArr(truckArr);
// this global friend function of Truck class will add a Truck instance, if there is available space
// in truckArr, else it will show an error message and return.
}
```

```
//show all trucks from truckArr
for(i=0; i<noOfTrucks; i++) cout<<truckArr[i]<<endl;
// the loop show information of ALL trucks
//define necessary friend function for above “cout<<truckArr[i]” statement
return 0;
}
```

**Q2: [20 marks]**

**Consider the following C++ code, and write the complete program (except main) for the given RUN.**

**You are not allowed to change the given main(). Also it is not required to write the main function in your**

**answer script.**

```
class Stack{
private: int *valPtr; int capacity, top;
public: //add necessary methods
//declare friends (if necessary)
};
int main(){
Stack stk1(10); //stack capacity is 10
stk1.push(30); stk1.push(25); stk1.push(15);

cout<<"All the popped elements of the stack are: "<<stk1<<endl;
cout<<"Average of the stack elements is: "<<stk1.getAvg()<<endl;
return 0;
}
```

RUN:

---

The capacity of the stack is set to 10

All the popped elements of the stack are: 15 25 30

Average of the stack elements is: 23.33

**Q3 : [20 marks]**

Consider the following C++ code, and write the complete program (except main) for the given RUN.

You are not allowed to change the given main(). Also it is not required to write the main function in your answer script.

```
class Array{
private: int *valPtr; int size;
public: //add necessary methods
};
int main(){
Array arr1, arr3;
arr1.setArray("Random",10); [2 marks for the
method]
arr2.setArray(3, "User input"); [2 marks for the method]
cout<<"Array arr1:"<<arr1<<endl; [3 marks for the
method]
cout<<"Array arr2:"<<arr2<<endl;
cout<<"Array arr3:"<<(arr1+arr2)<<endl; [3 marks for the method]
return 0;
}
```

RUN:

```
-------
How many ints? 4
Memory is allocated for 4 ints and Random values <10 are assigned to the array
Memory is allocated for 3 ints
Enter 3 integers for the array: 12 13 14
Array arr1: 1 2 7 9
Array arr2: 12 13 14
Array arr3: 1 2 7 9 12 13 14
```

**Q4 : [20 marks]**

**Consider the following C++ code, and write the complete program (except main) for the given RUN.**

**You are not allowed to change the given main(). Also it is not required to write the main function in your**

**answer script.**

```
class TwoDArray{
private: int **valPtr; int rows, cols;
public: //add necessary methods
};
int main(){
TwoDArray td1("Random",10), td2(2, 3, "User input");
cout<<"TwoDArray td1:"<<endl<<td1<<endl;
cout<<"TwoDArray td2:"<<endl<<td2<<endl;
return 0;
}
```

RUN:

```
-------
How many rows? 3
How many columns for 0-th row? 2
How many columns for 1-th row? 4
How many columns for 2-th row? 3
Memory is allocated for 3 rows and Random values <10 are assigned to the columns of those rows
Memory is allocated for 2 rows and 3 columns
Enter 3 integers for row-0: 12 13 14
Enter 3 integers for row-1: 15 16 17
TwoDArray td1:
```

1 2

2 5 3 7

5 1 8

TwoDArray td2:

12 13 14

15 16 17