BCS230
Lab10 Handout and Assignment

This lab handout provides four tasks that students are required to attempt during lab session. Lab objectives are for students to practice with pointers, classes, structures, and dynamic memory allocation.

Task1:

Run the following program and check its output.
Step1: Make grades a normal variable (not pointer) and then make all necessary changes for the code to work.
Step2: Add an int pointer data member to the structure called stdID before the courseNum and then use it in the main function similar to grades.
Step3: Create a dynamic array of 5 GradeList records, then using loops store 5 values, and print them.

```
#include <iostream>
using namespace std;

struct GradeList{
    string courseNum;
    int * grades;
};
int main (){
    int num=10;
    GradeList test1, *testPtr = &test1;
    (*testPtr).courseNum="BCS230";
    testPtr->grades=&num;
    cout<< testPtr->courseNum<<endl;
    cout<< *(*testPtr).grades <<endl;
    return 0;
}
```

Task2:

Write a program that dynamically allocates an array large enough to hold a user-defined number of test scores. Once all the scores are entered, the array should be passed to a function that sorts them in ascending order. Another function should be called that calculates the average score. The program should display the sorted list of scores and the averages with appropriate heading. Use pointer notation rather than array notation whenever possible. And don't accept a negative number for test scores.

Task3:

Run the following program and check its output.

Step1: Check what is the difference between returning an address of local variables to a function and returning parameter or dynamically allocated memory.

Step2: assign foo4() to p and then print p.

Step3: dereference p in step2

```
#include <iostream>
using namespace std;
int foo1(void){
    int p= 99;
    return p;
}

char *foo2(void){
    char buffer[] = "test_123";
    return buffer;
}

int *foo3(void){
    int t[3] = {1,2,3};
    return t;
}

int *foo4(void){
  cout<<"you just called me"<<endl;
    int *t;t=new int(5);
    return t;
}
int main(void)
{
    int *p;
    printf("foo1: %d\n", foo1());
    printf("foo2: %s\n", foo2());
    cout<<foo3()<<endl;
    cout<< " - > "<< foo4()<<endl;
}
```

Task4:

Run the following program and check its output.
Step1: Draw a diagram representation that resembles how student objects are linked to each other.
Step2: Traverse and print the students objects.
Step3: What kind of data-structure is in this code?

```
#include <iostream>
#include <vector>
using namespace std;
class Student{
private:
    int gpa;
    string name;
public:
    Student* next;
    Student(int g, string n, Student* nxt ){gpa=g;name=n; next= nxt;}
    string getName()      {   return name;        }
    int getGpa()          {    return gpa;          }
};

int main(){
   Student* nStd=nullptr;
   bool flg=false;
   string tmpName;int gpa;
   do{
    cout<<"Enter name: "; cin>>tmpName;
    cout<<"Enter GPA: " ; cin>> gpa;
    nStd=new Student(gpa, tmpName, nStd);
    cout<<"Do you want to continue (1 for YES/ 0 for NO) > ";cin>>flg;
   }while (flg);

  Student *ptrStd=nStd;

  ToDO: Step 2

  // clearing the memory is not discussed in this example
   return 0;
}
```

LAB10 ASSIGNMENT

Problem: download the attached datafile.txt. Read the file and then create a dynamic array that stores its elements.
Your code should use dynamic memory allocation, pointers, to find the sum and the average of all numbers.

- Should be completed before April 18th 2018 at 11:59 pm.
- Check rubric