ET580, Exam2, Spring 2023

Submission detail:

A) Submit cpp file for each question with given main function.
B) points distribution: 50% compilation, 50% correctness.

Q1: Implement a class called **Wallet** class as specified: (20 pt)
data members:

moneyList - a pointer of dynamic array of double.
size - size of moneyList.

functions:

Wallet(): default constructor set size to 0, moneyList to null pointer.
Wallet(int): one argument constructor set size, and create dynamic array use given size

accessor - an accessor for the **size** variable
operator[] - able to access or assign values to appropriate index

Note: if **moneyList** is not a pointer, 0 pt will be given to this question.

Use following main() to test your class.

```
int main(){
  Wallet a(5),b;
  for(int i=0;i<a.getSize();i++)
    a[i] = i*1.5;
  for(int i=0;i<a.getSize();i++)
    cout<<a[i]<<endl;
  cout<<endl;
}
```

Output from given main function:

```
0
1.5
3
4.5
6
```

Answer:

Q2: Continue with **Wallet** class: (20 pt)

Copy the previous answer to a new file, and implement following:

functions:

Update constructor like following:
void add(double): append given value to the moneyList
void spend(): remove the last value from moneyList
double total(): return the total value from moneyList
void output(): print all value of moneyList, also report the total amount in the wallet.

Use following main() to test your class.

```
int main(){
  Wallet a(2),b;
  a[0] = 0.25;
  a[1] = 1;
  a.output();
  a.add(20);
  a.add(10);
  a.output();
  cout<<a.total()<<endl;
  for(int i=0;i<6;i++)
    a.spend();
  a.output();
}
```

Output from given main function:

```
Money in wallet:
$0.25
$1
Total: 1.25
Money in wallet:
$0.25
$1
$20
$10
Total: 31.25
31.25
No more money to spend.
No more money to spend.
Money in wallet:
Total: 0
```

Answer:

Q3: Continue with **Wallet** class: (20 pt)
big three: Implement big three for **Wallet** class.
Add notification message for each of big three.

Use following main() to test your class.

```
int main(){
  Wallet a(2),c;
  a[0] = 0.25;
  a[1] = 1;
  Wallet b = a;
  a.add(20);
  a.output();
  b.output();
  c = a;
  a.add(30);
  a.output();
  c.output();
}
```

Output from given main function:

```
==> Copy constructor is called
Money in wallet:
$0.25
$1
$20
Total: 21.25
Money in wallet:
$0.25
$1
Total: 1.25
==> Assignment operator is called
Money in wallet:
$0.25
$1
$20
$30
Total: 51.25
Money in wallet:
$0.25
$1
$20
Total: 21.25
==> Destructor is called
==> Destructor is called
==> Destructor is called
```

Answer:

Q4: Create following functions, and test use given main function. (20pt)

int create2DArray(int row, int column)**: return a 2d dynamic int array with given row and column.
void populate(int array, int row, int column, int start)**: initialize the given dynamic 2d array use number from start, and increase one every element.
void print(int array, int row, int column)**: print every element from given dynamic 2d array.
void delete2DArray(**int** array, int row, int column): which delete given dynamic 2d array from the heap.**

Use pointer arithmetic for every function.

Use following main() to test your class.

```
int main(){
  int **a = create2DArray(3,5);
  populate(a,3,5,10);
  print(a,3,5);
  delete2DArray(a,3,5);
  cout<<endl;
  a = create2DArray(2,6);
  populate(a,2,6,22);
  print(a,2,6);
  delete2DArray(a,2,6);
}
```

Output from given main function:

```
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24

22 23 24 25 26 27
28 29 30 31 32 33
```

Answer:

Q5: Write a function for given main function (20pt)

void bargraph(int n)

which takes an integer n and print the bar graph based on the digit of given integer. Use **pointer arithmetic** and **Dynamic Array** if you would like to use array.

Use following main() to test your class.

```
int main(){
  bargraph(12945);
  bargraph(6789102);
}
```

Output from given main function:

```
    X
    X
    X
    X
    X   X
    X X X
    X X X
  X X X X
X X X X X
----------
1 2 9 4 5

      X
    X X
  X X X
X X X X
X X X X
X X X X
X X X X
X X X X     X
X X X X X   X
--------------
6 7 8 9 1 0 2
```

Answer: