2160 Lab 2

Patrick Taylor and James Wachala

Section Number: 1

January 1st, 2019

Pre-Lab b)

```
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 1
You selected "Print the array"
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 3
You selected "Remove last element"
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 4
You selected "Insert one element"
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
```

```
4. Insert one element
5. Exit

Select an option: 5
You selected "Exit"
```

1: The Grow() function code

```
void Grow() {
    // creates new pointer to point to the larger array
    size *= 2;
    double *nv = new double[size];
    std::cout << "created nv \n";
    // copies old values into new array
    for (int i = 0; i < size / 2; i++) {
        nv[i] = v[i];
        std::cout << "cp\n";
    }
    // makes v point to new array
    delete[] v;
    v = nv;
    std::cout << "Vector grown\nPrevious capacity: " << size / 2
              << " elements\nNew capacity: " << size << " elements\n";
}
```

2A: The AddElement() and PrintVector() function code

```
void AddElement() { // COMPLETE
    if (count < size) {
        double usrNum;
        std::cout << "Enter the new element: ";
        std::cin >> usrNum;
        std::cout << '\n';
        v[count] = usrNum;
        count++;
    } else {
        Grow();
        AddElement();
    }
}

void PrintVector() { // COMPLETE
    for (int i = 0; i < count; i++) {
        std::cout << v[i] << " ";
    }
    std::cout << "\n";
}
```

2B: Output (when doing AddVector() to the point the vector grows)

```
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Enter the new element: 1

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Enter the new element: 2

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Vector grown
Previous capacity: 2 elements
New capacity: 4 elements
Enter the new element: 3

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Enter the new element: 5

Main Menu:
1. Print the array
2. Append element at the end
```

```
3. Remove last element
4. Insert one element
5. Exit

Select an option: 1
You selected "Print the array"
1 2 3 5
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option:
```

3A: RemoveElement() function code

```
void RemoveElement() { // COMPLETE
    if (debug == 1) {
        std::cout << "Count: " << count;
        std::cout << "\nSize: " << size << '\n';
        // std::cout << "Size:" << size;
    }
    if (count == 0) {
        std::cout << "Error: No values in the Array." << '\n';
        std::cout << "Returning to Main Menu..."
                  << "\n\n";
    }

    else {
        count--;
    }
    if (debug == 1) {
        std::cout << "Count: " << count;
        std::cout << "\nSize: " << size << '\n';
        // std::cout << "Size:" << size;
    }
    if((count/1.0)/size)
        Shrink();
}
```

3B: Output (A, removing last element succesfully. B, removing on an empty vector)

```
Main Menu:
1. Print the array
2. Append element at the end
```

```
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Enter the new element: 54

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 3
You selected "Remove last element"
Count: 1
Size: 2
Count: 0
Size: 2
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 3
You selected "Remove last element"
Count: 0
Size: 2
Error: No values in the Array.
Returning to Main Menu...

Count: 0
Size: 2
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option:
```

4A: InsertElement() function code

```
void InsertElement() { // NOT COMPLETE
    int index;
    double usrNum;
    if (debug == 1) {
        std::cout << "Count: " << count;
        std::cout << "\nSize: " << size << '\n';
        // std::cout << "Size:" << size;
    }
    std::cout << "Enter the index of the new element: ";
    std::cin >> index;
    if (index > count) {
        std::cout << "Na fam, u wrong \n";
        InsertElement();
    }
    std::cout << "Enter the new element: ";
    std::cin >> usrNum;
    if (count == size) {
        Grow();
    }
    for (int i = count - 1; i >= index; i--) {
        v[i + 1] = v[i];
    }
    v[index] = usrNum;
    count++;
}
```

4B: Output (A, Adding several elements. B, entering an invalid value for the index)

```

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Enter the new element: 34

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 4
You selected "Insert one element"
Count: 1
Size: 2
Enter the index of the new element: 54
Na fam, u wrong
Count: 1
```

```
Size: 2
Enter the index of the new element: 0
Enter the new element: 12
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Vector grown
Previous capacity: 2 elements
New capacity: 4 elements
Enter the new element: 56

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 4
You selected "Insert one element"
Count: 3
Size: 4
Enter the index of the new element: 2
Enter the new element: 13
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 1
You selected "Print the array"
12 34 13 56
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option:
```

Extra Credit: Shrink() function code

```
void Shrink() { // COMPLETE
    // creates new pointer to point to the smaller array
    std::cout<<"Vector Utilization below " << ((count/1.0)/size)*100 <<
    "\%. Deleting half of the vector (only empty space).";
    size /= 2;
    double *nv = new double[size];
    std::cout << "created nv \n";
    // copies old values into new array
    for (int i = 0; i < size / 2; i++) {
        nv[i] = v[i];
        std::cout << "cp\n";
    }
    // makes v point to new array
    delete[] v;
    v = nv;
    std::cout << "Vector shrunk\nPrevious capacity: " << size * 2
              << " elements\nNew capacity: " << size << " elements\n";
}
```

Extra Credit: Output when vector is shrunk

The function outputs at what percentage the vector was utilized, but will only run Shrink() if the utilization is below 30%.

```

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Enter the new element: 45

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Enter the new element: 46

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
```

```
4. Insert one element
5. Exit

Select an option: 2
You selected "Append element at the end"
Vector grown
Previous capacity: 2 elements
New capacity: 4 elements
Enter the new element: 47

Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 1
You selected "Print the array"
45 46 47
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 3
You selected "Remove last element"
Count: 3
Size: 4
Count: 2
Size: 4
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
4. Insert one element
5. Exit

Select an option: 3
You selected "Remove last element"
Count: 2
Size: 4
Count: 1
Size: 4
Vector Utilization below 25%. Deleting half of the vector (only empty
space).Vector shrunk
Previous capacity: 4 elements
New capacity: 2 elements
Main Menu:
1. Print the array
2. Append element at the end
3. Remove last element
```

```
4. Insert one element
5. Exit

Select an option:
```