**CPSC 1020 SPRING 2016**

**EXAM #3**

**This is a Closed Book exam. Please keep your notes and your computers closed. You have 50 minutes to complete the Exam. There is a total of 110 points 100 regular points and 10 bonus points.**

**Question 1: (42 points)**

**Write the implementation for MyInteger class. I have provided a sample main with output.**

class MyInteger

{

public:

MyInteger();

MyInteger(int newint);

void setInt(int newint);

int getInt();

int operator[](int index);

private:

int num;

};

**3 points**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Defalut constructor \*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**3 points**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Constructor to initialize data \*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/**

**3 points**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Setter \*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**3 points**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Getter \*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**30 points**

**Please read carefully!**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Overrides [] operator \*

\*This operator returns the digit in position i, where i = 0 is the least-significant \*

\*digit. Ex. 123 the least-significant digit is 3. \*

\*If no digit exist then -1 should be returned. Ex. the array is out of bounds \*

\*If the parameter passed in is negative, print a message that states the index \*

\* was an illegal value and return -1. \*

\*As an example, if X is of type MyInteger and is set to 418, then x[0] \*

\*should return 8, X[1] should return 1, X[2] should return 4, X[3] should \*

\*return -1 and x[-8] should print Illegal value and should return -1. \*

\***Hint**: you will use the modulus and division operators \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* main function

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main(){

MyInteger num(418);

cout << num[0] << " " << num[1] << " " << num[2] << endl;

cout << num[3] << endl;

**Output of this program:**

**8 1 4**

**-1**

**Illegal index value.**

**-1**

cout << num[-8] << endl;

return 0;

}

**QUESTION 2 (25 points)**

**What will this program output? Read carefully and give all output.**

#include <iostream>

using namespace std;

void recursiveReverse( char a[], int start, int end ){

if ( start >= end ) // job is done!

return;

if ( start + 1 == end ) {

char temp = a[start]; // only two elements, swap them

a[start] = a[end];

a[end] = temp;

return;

}

char temp = a[start];

a[start] = a[end];

a[end] = temp;

recursiveReverse( a, start + 1, end - 1); // make recursive call to finish

}

int main(){

char a[20];

int i;

for ( i = 0; i < 20; i++ )

a[i] = 'A' + i;

cout << "array: " << endl;

for ( i = 0; i < 20; i++ )

cout << a[i] << " ";

cout << endl;

recursiveReverse( a, 5, 19);

cout << "array with elements with index 5-19 reversed" << endl;

for ( i = 0; i < 20; i++ )

cout << a[i] << " ";

Output:

cout << endl;

return 0;

}

**Question 3: (13 points)**

In class we discussed three types of sorting algorithms.

Name the three:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now in words describe the quick sort algorithm:

**Question 4: (10 points) Dynamic Memory Allocation**

Given the following pointer declaration, dynamically allocate memory for this pointer.

double \*dPtr:

When your program is done with this memory you should return the memory to the operating system. Write the code to return the memory you allocated above to the operating system.

**Question 5: (10 points) Member Initialization List**

**Consider the code below and complete the Person constructor using the member initialization list technique.**

class Date{

private:

string month;

int day, year;

public:

Date(string m, int d, int y){month = m; day = d; year = y}

Date(){month = " "; day = 0; year = 0;}

void setMonth(string m){month = m;}

string getMonth() { return month; }

};

**/\*Write the constructor below using member initialization list to initialize the Date and Person class member variables Note: this is NOT an inheritance problem. Person “has a” Date. \*/**

class Person

{

private:

string name;

Date dateOfBirth;

public:

**THIS IS THE CONSTRUCTOR YOU ARE TO COMPLETE**

Person(string name, string month, int day, int year)

string getMonth() { return dateOfBirth.getMonth();}

};

**Bonus Question (5 points)**

What is a dangling pointer and how can you fix it:

What causes a memory leak and how can you fix it:

**Bonus Question: (5 points)**

The code below does not actually swap the values of a and b. Using C++ **PASS BY REFERENCE (NOT A POINTER)** fix the code so that it will swap the values of a and b correctly.

**Hint: this basically only involves changing two lines of code.**

#include <iostream>

using namespace std;

// function declaration

void swap(int x, int y);

int main ()

{

// local variable declaration:

int a = 100;

int b = 200;

cout << "Before swap, value of a :" << a << endl;

cout << "Before swap, value of b :" << b << endl;

/\* calling a function to swap the values.\*/

swap(a, b);

cout << "After swap, value of a :" << a << endl;

cout << "After swap, value of b :" << b << endl;

return 0;

}

// function definition to swap the values.

void swap(int x, int y)

{

int temp;

temp = x;

x = y;

y = temp;

return;

}