**Çankaya University**

**Department of Computer Engineering**

**CENG 241 Advanced Programming Final**

### Instructor: Abdul Kadir GORUR

Time Allowed: **115 minutes** Date: 24/ 11 / 2014

Name and Surname : ………………………………...........

Student Number : ……........……………………………

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Q1(12p)** | **Q2(38p)** | **Q3(12p)** | **Q4(8p)** | **Q5(10p)** | **Q6(20p)** | **Bonus(5p)** | **Total(105p)** |
|  |  |  |  |  |  |  |  |

**The principle of least privilege must be satisfied in all questions with the proper use of const qualifier whenever it is required. (there will be -1p (minus one point) penalty for each missing const)**

**Q1)(12p)** Inventory BinsWrite a program that simulates inventory bins in a warehouse. Each bin holds a number of the same type of parts. The program should use objects of class Bin types that keeps the following data:

**Description of the part kept in the bin**

**Number of parts in the bin**

The program should have an array of 5 bins, initialized with the following data: (7pts for class design, 3pts for main program)

|  |  |  |
| --- | --- | --- |
| **Bin No** | **Part Description** | **Number of Parts in Bin** |
| 0 | Valve | 10 |
| 1 | Flange | 7 |
| 2 | Gear | 5 |
| 3 | Gear housing | 5 |
| 4 | Cable | 18 |

**Q2)(34p+2p+2p)** Number Array Class  
Design a class that has an array of floating-point numbers. The constructor should accept an integer argument and dynamically allocate the array to hold that many numbers **(2p).** Default constructor should set size to 10. Once the array is formed its size will be fixed and will not change during the life time of object instantiated from NumberArray class. (your design should handle all error conditions)

The destructor should free the memory held by the array **(2p)**. In addition, there should be member functions to perform the following operations:

* Store a number in any element of the array (3p)
* Retrieve a number from any element of the array(2p)
* Return the highest value stored in the array(3p)
* overload [] operator(5p)
* overload << operator which will print all the elements stored in object(5p)
* copy constructor(5p)
* overloaded = operator (5p)
* overloaded == operator, which will return true if both NumberArray objects are clone of each other. (6p)

class NumberArray

{

private:

double \*ptr;

int size;

**Q3) (12p)** RetailItem Class Write a class named RetailItem that holds data about an item in a retail store. The class should have the following member variables:

• description . A string that holds a brief description of the item.

• unitsOnHand . An int that holds the number of units currently in inventory.

• price . A double that holds the item’s retail price.

Write a **constructor** that accepts arguments for each member variable, appropriate **mutator** functions that store values in these member variables, and **accessor** functions that return the values in these member variables. Once you have written the class, write a main function that creates three RetailItem objects and stores the following data in them.

|  |  |  |
| --- | --- | --- |
| **Description** | **Units On Hand** | **Price** |
| Rotring | 5 | 20.22 |
| Pelikan Fountain Pen | 10 | 255.60 |
| Moleskine Classic Notebook | 7 | 35.5 |

Q4**)(8p)** What is printed by this program?Two point for each printf *Answer in the box*:

|  |  |
| --- | --- |
|  | **Address** |
| **C** | **0xAFFC2** |
| **E** | **0xAFFC3** |
| **N** | **0xAFFC4** |
| **G** | **0xAFFC5** |
| **2** | **0xAFFC6** |
| **4** | **0xAFFC7** |
| **1** | **0xAFFC8** |
| **'\0'** | **0xAFFC9** |

#**include <stdio.h>**

**void main()**

**{**

**char \*p = “CENG241”;**

**printf("%c\n", \*(p + \*(p+2) – 'P' + 2));**

**printf("%c\n", \*(p + 4));**

**printf("%c\n", \*p);**

**printf("%p\n", (p+\*p –'A' +1 ));**

**}**

Q5)**(10p)** Write a function called indexOf, that takes two c-type strings as its arguments. This function will search first string in second one, if it exists, it will return starting position of first string within second one, otherwise function returns -1.

(you are not allowed to use C++ string class, c-type string library functions)

Ex:

indexOf("ing", "programming in C++ is full of fun"); will return 7

int indexOf(char str[], char s[])

{

int i, j;

i = 0;

while (s[i] != '\0')

{

j = 0;

while (str[j] != '\0' && s[i + j] != '\0')

{

if (str[j] != s[i + j])

break;

j++;

}

if (str[j] == '\0')

return i;

i++;

}

return -1;

}

**Bonus Question (5p)**

Complete the following main function by writing minimum number statement, that is supposed to find total number of characters in **str** array. You are not allowed to use any library function.

void main()

{

char str[4][10]={"yilmaz", "nesib", "murat", "arda" };

}

**Q6)(20p)** Numbers Class Design a class Numbers that can be used to translate whole dollar amounts in the range 0 through 9999 into Turkish description of the number. For example, the number 713 would be translated into the string **yedi yuz on uc**, and 8203 would be translated into:  
 **sekiz bin iki yuz uc**.

The class should have a single integer member variable:

**int number;**

and a static array of string objects that specify how to translate amounts into the desired format.

The class should have a constructor that accepts a nonnegative integer and uses it to initialize the Numbers object. It should have a member function print() that prints the Turkish description of the Numbers object. Demonstrate the class by writing a main program that asks the user to enter a number in the **proper range** and then prints out its Turkish description.

#include <iostream>

#include <string>

using namespace std;

class Numbers{

private:

int number;

//declarations for static strings

static string birler[9];

static string onlar[9];

static string yuzler;

static string binler;

public:

Numbers(int n=0)

{

if ( n < 0 || n > 9999 )

exit(0);

number = n;

}

void print()const

{

cout << convert() << endl;

}

private:

string getDigitName(int digit)const

{

string res("");

if( digit > 0 )

if (digit > 1)

res = birler[digit - 1];

return res;

}

string convert()const

{

int n = number;

int digit;

string result;

digit = n / 1000;

if (digit > 0)

result = getDigitName(digit) + " " + binler;

n = n % 1000;

digit = n / 100;

if (digit > 0)

result += " "+getDigitName(digit) + " " + yuzler;

n = n % 100;

digit = n / 10;

if (digit > 0)

result += " " + onlar[digit - 1];

n = n % 10;

digit = n / 1;

if (digit > 0)

result += " " + birler[digit - 1];

return result;

}

};

//definition and initializations fop static private members of Numbers class

string Numbers::birler[9] = {"bir","iki","uc","dort","bes","alti","yedi","sekiz","dokuz" };

string Numbers::onlar[9]={"on","yirmi","otuz","kirk","elli","atmis","yetmis","seksen", "doksan" };

string Numbers::yuzler = "yuz";

string Numbers::binler = "bin";

void main()

{

int n = 5205;

Numbers obj(n);

obj.print();

}