|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | | **Course:** | **Computer Programming Lab** | **Course Code:** | **CS-103** | |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Fall 2017** | |
| **Duration:** | **120 Minutes** | **Total Marks:** | **30** | |
| **Paper Date:** | **7-Dec-2017** | **Weight** | **-** | |
| **Section:** | **A,B,C** | **Page(s):** | **7** | |
| **Exam:** | **Final Term** | **Roll. No.** |  | |
|  | |  | | | | |

**Instructions:**

1. Understanding the question paper is also part of the exam, so do not ask any clarification.
2. Make sure to switch off your mobile phones before the Exam starts.
3. No USB’s are allowed. No INTERNET is allowed. Please see that the area in your threshold is clean. You will be charged for any material which can be classified as ‘helping in the paper’ found near you.
4. Talking/Discussion is not allowed. It is your responsibility to protect your code and save it from being copied. If you don’t protect it all matching codes are considered copy/cheating cases.

**Question 1**

**You are given partial definition of class Date and Person and driver program. Your task is to update class definitions such that driver program runs successfully.**

**class Date {**

**private:**

intyear; // 1753-9999

intmonth; // 1-12

intday; // 1-31

char \* dateString; // Monday, 9 Mar 2017

const static char STR\_MONTHS[12][4];

const static char STR\_DAYS[12][10];

const static int DAYS\_IN\_MONTHS[12];

const static int YEAR\_MIN = 1753;

const static int YEAR\_MAX = 9999;

**public:**

static bool Compare(Date date1,Date date2){ if(date1.year>date2.year) return true; return false;}//returns true if date1 is greater than date2

intgetDayOfWeek(int y, int m, int d);

Date(int y = 1, int m = 1, int d = 2000); //Constructor with default arguments

void setDateString(int y, int m, int d); //Initialize the char \* date with exact dateString.

void print();

**};**

// Initialize static non-integer variable (must be done outside the class declaration)

const char Date::STR\_MONTHS[12][4] = { "Jan", "Feb", "Mar", "Apr", "May", "Jun",

"Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

constintDate::DAYS\_IN\_MONTHS[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

const char Date::STR\_DAYS[12][10] = { "Sunday", "Monday", "Tuesday", "Wednesday",

"Thursday", "Friday", "Saturday" };

// A static function that returns the day of the week (0 means Sun, 6 means Sat) for the given date

**intDate::getDayOfWeek(int y, int m, int d)**

**{**

intcenturyTable[] = {4, 2, 0, 6, 4, 2, 0, 6}; // 17xx, 18xx, ...

intMonthTable[] = {0, 3, 3, 6, 1, 4, 6, 2, 5, 0, 3, 5};

intMonthLeapYearTable[] = {6, 2, 3, 6, 1, 4, 6, 2, 5, 0, 3, 5};

int century = y / 100;

inttwoDigitYear = y % 100;

intcenturyTableIndex = (century - 17) % 8;

// Date before 17xx are not valid, but needed to prevent negative index

if (centuryTableIndex< 0)

{

centuryTableIndex += 8;

}

int sum = centuryTable[centuryTableIndex] + twoDigitYear + twoDigitYear / 4;

if (isLeapYear(y))

{

sum += MonthLeapYearTable[m-1];

}

else

{

sum += MonthTable[m-1];

}

sum += d;

return sum % 7;

**}**

**Int main()**

**{**

Date bday(1965, 10, 15);

Date deathDay(2005, 7, 20);**//Death Date is greater than birth Date and less than Current Date so this should create the object**

If(Date::ValidateDate(deathday,bday) == true)

{

//deathDay should be greater than birthday and less than currentDate

//If deathDay is valid then create and print the object otherwise print error message

Person p1(“Ahsan Naeem”, bday, deathDay); //Write Person class yourself

P1.Print();

// Ahsan Naeem

// Date Of Birth: Friday, 15 October, 1965

// Death Date: Wednesday, 20 July, 2005

}

Else

{

// cout<< Death Day is Invalid

}

Bday.SetDate(1965, 10, 15);

deathDay.SetDate(1955, 7, 20);**//Death Date is less than birth Date so this should not create the object**

If(Date::ValidateDate(deathday,bday) == true )

{

//deathDay should be greater than birthday and less than currentDate

//If deathDay is valid then create and print the object otherwise print error message

Person p1(“Ahsan Naeem”, bday, deathDay);

P1.Print();

// Ahsan Naeem

// Date Of Birth: Friday, 15 October, 1965

// Death Date: Wednesday, 20 July, 2005

}

Else

{

// cout<< Death Day is Invalid

}

Bday.SetDate(1965, 10, 15);

deathDay.SetDate(2018, 7, 20);**//Death Date is greater than current Date so this should not create the object**

If(Date::ValidateDate(deathday,bday) == true )

{

//deathDay should be greater than birthday and less than currentDate

//If deathDay is valid then create and print the object otherwise print error message

Person p1(“Ahsan Naeem”, bday, deathDay);

P1.Print();

// Ahsan Naeem

// Date Of Birth: Friday, 15 October, 1965

// Death Date: Wednesday, 20 July, 2005

}

Else

{

// cout<< Death Day is Invalid

}

return 0;

}

**Question 2**

**We have a class *PrintedMedia*which have three child classes *Book, Journal and Magazine.* All kind of media items have their Issue date .**

Book Class has, AuthorName(String), Edition(int).

Journal Class has, PublisherName(String), ImpactFactor(float).

Magazine has Name(String), Category(int)

Media Items (Books, Journal, Magazines are categorized in 1,2,3 respectively.). We are given with a file, named “mediaItems.txt” which has Number of media items on its first line. After 1st line, on every line first value is type of media Item. Next values are of date of issuance will be in “day” “month” “year”, space separately.

In the case it is 1(Book), Next value is Edition of book, then the Name of the book.

In the case of 2(Journal), Next Value is Impact Factor and then Publisher of Journal.

In the case of 3(Magazine), Next Value is Category and then Name of Magazine. Category for Magazines are (1-Kids, 2-Fashion, 3-Sports.)

**File Format:**

|  |
| --- |
| <Total number of media files>  <Category of media item><Issueday><Issuemonth><IssueYear><Book Edition><Book Name>  <Category of media item><Issueday><Issuemonth><IssueYear><Impact Factor><Publisher Name>  <Category of media item><Issueday><Issuemonth><IssueYear><Category><Name> |

Your task is to write code such that is prints Required Output. Partial code of main function is given below.

**Void main()**

**{**

//read the number of Media Items.

//Initialize the Printed Media array.

//Read the data from file and fill the array with data.

For(inti=0; i<NumberOfMediaItems; i++)

Arr[i]->Print();

For(inti=0; i<NumberOfMediaItems; i++)

Delete Arr[i];

}

**mediaItems.txt** contains following data:

|  |
| --- |
| 5 //Total number of print media items  1 20 12 2015 3 Data Structures  2 15 10 2012 6.5 Semantics Journal  1 20 12 2015 6C plus plus Programming  3 1 5 2016 3 Sports Latest  3 1 5 2016 1 Readers |

**Output Required:**

|  |
| --- |
| Book: Data Structures  Issue Date: Monday 20 Dec 2015**//Use Date class of Question 1**  Edition: 3  Journal: Semantics Journal  Issue Date: Monday 15Oct 2012  Impact Factor: 6.5  Book: C plus plus Programming  Issue Date: Monday 20 Dec 2015  Edition: 6  Magazine: Sports Latest  Issue Date: Monday 1May 2016  Category: Sports  Magazine: Readers  Issue Date: Monday 1 May 2016  Category: Kids |

**Question 3**

Along with basic information (Name, Edition Issued Date) a Book may save BookText. Book Text contains **TableOfContent** , **Apendix**, **DetailedContent**.

**TableOfContent** Class has an array of Page Numbers, and 2D char array which will hold the headings of topics. You will have to read the data from file “TableOfContent.txt” in which on first line there will be the number of records. And then on the every line you will have pageNumber and Heading space separately.

**TableOfContent.txt** contains following data:

|  |
| --- |
| 5  1 Arrays  22 Pointers  54 Structures  98 Classes and Aggregation  155 Inheritance and Poly morphism |

**ApendixClass** have an array of 2D char array which will hold the words and array of Page Numbers (int). You will have to read the data from file “Apendix.txt” in which on first line there will be the number of records. And then on the every line you will have pageNumber and a keyword separated by a space.

**Apendix.txt** contains following data:

|  |
| --- |
| 3  31 index  2 ptr  67 line |

**DetailedContent** class will have only a char array (of 300 characters) which can hold whole content of book.

**DetailedContent.txt** contains following data:

|  |
| --- |
| The values of the members tm\_wday and tm\_yday of *timeptr* are ignored, and the values of the other members are interpreted even if out of their valid ranges (see [struct tm](http://www.cplusplus.com/tm)). For example, tm\_mday may contain values above 31, which are interpreted accordingly as the days that follow the last day of the selected month. |

**Your task is to update the code of Question 2 such that it prints following output:**

**Output Required:**

|  |
| --- |
| Book: Data Structures  Issue Date: Monday 20 Dec 2015**//Use Date class of Question 1**  Edition: 3  **Book Text:**  **Table Of Contents:**  1 Arrays  22 Pointers  54 Structures  98 Classes and Aggregation  155 Inheritance and Poly morphism  **Detailed Content:**  The values of the members tm\_wday and tm\_yday of *timeptr* are ignored, and the values of the other members are interpreted even if out of their valid ranges (see [struct tm](http://www.cplusplus.com/tm)). For example, tm\_mday may contain values above 31, which are interpreted accordingly as the days that follow the last day of the selected month.  **Index:**  31 index  2 ptr  67 line  Journal: Semantics Journal  Issue Date: Monday 15Oct 2012  Impact Factor: 6.5  Book: C plus plus Programming  Issue Date: Monday 20 Dec 2015  Edition: 6  **Book Text: Not Available**  Magazine: Sports Latest  Issue Date: Monday 1May 2016  Category: Sports  Magazine: Readers  Issue Date: Monday 1 May 2016  Category: Kids |