|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** | **Course Code:** | **CS-103** |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Fall 2016** |
| **Duration:** | **120 Minutes** | **Total Marks:** | **35** |
| **Paper Date:** | **20-Oct-16** | **Weight** | **-** |
| **Section:** | **A,B** | **Page(s):** | **4** |
| **Exam:** | **Midterm** | **Reg. No.** |  |

**Instruction/Notes:**

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. 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.
5. You are not allowed to use internet for any purpose.

**Submission Path:**

[\\xeon\Fall2016\Ahmad Raza\CP-Lab Mid\Q2](file:///\\xeon\Fall2016\Ahmad%20Raza\CP-Lab%20Mid\Q2) for Question 2 (Format: 15L-1234\_Q1.cpp)

[\\xeon\Fall2016\Ahmad Raza\CP-Lab Mid\Q3](file:///\\xeon\Fall2016\Ahmad%20Raza\CP-Lab%20Mid\Q3) for Question 3 (Format: 15L-1234\_Q2.cpp)

**Q. No. 1:**

**Considering the given piece of code below, write the output of each of the cout statement. [5 + 10 Marks]**

int r=4, c=3;

int \*\*ar;

ar=new int\*[r];

for (int i=0;i<r;i++)

ar[i]=new int [c];

for (int i=0;i<r;i++)

for (int j=0;j<c;j++)

ar[i][j]= i+j;

|  |  |
| --- | --- |
| **Output Statement** | **Output (as on Screen)** |
| cout << \*(&ar[0][0]+2); |  |
| cout << \*(&\*\*ar +2); |  |
| cout << \*&ar[2][2] ; |  |
| cout << \*(\*(ar + 1) + 2) + \*\*ar + 1 ; |  |
| cout << \*\*(\*(\*(ar + 0) + 1) + &\*ar) ; |  |

|  |  |
| --- | --- |
| Code | Output |
| #include<iostream>  using namespace std;  class Point  {  private:  int x;  int y;  public:  Point(int a ,int b = 20)  {  x = a;  y = b;  cout << "Per con " << x << " " << y << "\n";  }  Point()  {  cout << "Default con " << x << y << "\n";  }  Point(Point & p)  {  x = p.x;  y = p.y;  cout << "Copy con " << x << " " << y << "\n";  }  Point getGreater(Point p)  {  if(x > p.x && y > p.y ) return \*this;  else return p;  }  Point getSmaller(Point &p)  {  if(x < p.x && y < p.y ) return \*this;  else return p;  }  Point &graterSmaller(Point p)  {  if(x > p.x && y < p.y ) return \*this;  else return p;  }  Point &smallerGreater(Point &p)  {  if(x < p.x && y > p.y ) return \*this;  else return p;  }  ~Point()  {  cout << "dest " << x << " " << y << "\n";  }  };  int main()  {  Point P;  Point P1(1);    Point P2(1,2);  Point P3 = Point(5,6);    Point P4 (3,4);    Point P5 = P2.getGreater(P4);  P2.getSmaller(P4);  Point P6 = P2.graterSmaller(P4);  P2.smallerGreater(P4);  return 0;  } |  |

**Q. No. 2: [10 Marks]**

A Common place to buy candy is from a candy machine. A new candy machine has been bought for the gym, but is not working properly. The machine sells candies, chips, gum, and cookies. You have been asked to write a class for this **candy machine** so that it can be put into operation.

The program should do the following:

1. Show the customer the different products sold by the candy machine and their prices. (**Note this data would be taken from an input.txt file**).
2. Let the customer make selection of products and required number of items. If the selected product is not available program should prompt the appropriate message and ask the customer to choose another product.
3. Show the total bill of selected items.
4. Accept money from customer until user enter the require amount to buy the selected products.
5. Release the items. And update the text file by the remaining number of items of every product.

Write a menu driven C++ program.

**For Example:**

Input.txt

Product Price of one item Number of Items Available

Candies 1 50

Chips 20 5

Gums 5 25

Cookies 10 14

If customer purchases 10 candies, 5 chips and 3 cookies then input.txt file should be updated as

Product Price of one item Number of Items Available

Candies 1 40

Chips 20 0

Gums 5 25

Cookies 10 11

**Q. No. 3: [10 Marks]**

Use following code and implement missing functions. We want to implement “Test” class in such a way that each object will contains a dynamic array with size of array, and “CalculateFactor” will calculate factor of each element in an array of that object that will call “CalculateFactor” and save factors in a dynamic 2-D array.

Note that column size of each row should be equal to number of factors. Don’t create columns with same size.

You are allowed to use any logic to implement this but don’t change code in “main()”.

#include<iostream>

using namespace std;

class Test

{

private:

int \*ptr;

int\* size;

public:

//Write Constructors, Functions…

};

int main()

{

int arr[4]={2,3,4,5};

Test t1(arr,4);

Test t2(arr);

Test t3(t2);

t1.Print();

cout<<endl;

t2.Print();

cout<<endl;

t3.Print();

int\*\* ptr=t1.CalculateFactor();

ptr.PrintFactors();

system("pause");

return 0;

}

**Note about Factor:**

Factors of 12 are 1,2,3,4,6,12

Factors of 10 are 1, 2,5,10