جامعة سيناء
Sinai University

Faculty of Information Technology

Final Exam of Computer Programming 2 (CSW 234)

Academic Year: 2015/2016
Level: 2nd Level | **Semester: 1st Semester**
Date: 9-1-2016 | **Time: 3 hours**

Answer about the following questions **(Total: 60 marks)**

Question 1: Show the output of the following program: **[15 Marks]**

```
class Circle {
   protected:
      int x, y, r;
   public:
     Circle(int xVal, int yVal, int rVal)
      { x=xVal;  y=yVal; r=rVal;}
      virtual void display()
      {cout << x << “,“ << y << "," << r; }
      float getArea() { return 3.14 * r * r; }
};
class Cylinder  : public Circle {
   private:
      int height;
   public:
      Cylinder(int xVal, int yVal, int rVal, int h)
                      : Circle(xVal, yVal, rVal)
      { height = h; }
      void display()
      {  cout << x << “,“  << y << ","
            << r << “,“ << height; }
      float getArea()
      { return 2 * 3.14 * r * height; }
};

void main () {
  Circle* shape[4];
  shape[0]=new Circle(3, 5, 10);
  shape[1]=new Cylinder(6, 8, 5, 20);
  shape[2]=new Circle(7, 9, 15);
  shape[3]=new Cylinder(1, 3, 7, 5);
  for (int i=0; i<4;i++)
  {
    shape[i]->display();
    cout<<"\nArea:" <<shape[i]->getArea();
  }
}
```

Question 2: **[15 Marks]**

(a) Define a class ***counter***, which has three data members: ***count*** (of type int) that represents the value of the counter, ***overflow*** (of type bool) that is set to *true* if an overflow occurs, and ***maxcnt*** (of type int) that represents the maximum value of the counter. It has a constructor ***counter(int max_count)***, that creates a counter with a given maximum count and an initial value 0, and sets *overflow* to *false*, and it has 4 member functions: ***increment()***, which increases the current count by 1; but if the current count equals the maximum count, the *count* is set to zero, and *overflow* to true; ***reset()***, which sets the counter to zero and overflow to false; ***get_count()***, which returns the current count, and ***check_for_overflow()***, which returns true if an overflow has occurred and false if it has not.

(b) Write a main program that reads a sentence, and counts the frequency of each vowel (a, e, i, o, u) in it, then displays these frequencies. The program should use a separate counter for each vowel, and it should display an error message if any of the counters overflows

Question 3: **[15 Marks]**

(a) Declare and implement a class ***fraction*** that has two data members: ***num*** and ***denom***, (both of type **int**), which represent the fraction's numerator and denominator, respectively. It has two member functions: ***getFract()*** that reads the ***num*** and ***denom*** of a ***fraction*** object from the keyboard in fractional form (i.e. num/denom), and ***showFract()*** that displays a ***fraction*** object in fractional form. It also has two overloaded operators **+** and *****, for performing the addition and multiplication operations on any two objects of class ***fraction***, respectively. Note that, the formulas for adding and multiplying two fractions a/b and c/d are as follows:

$$\frac{a}{b}+\frac{c}{d}=\frac{a*d+b*c}{b*d}, \qquad \frac{a}{b}\times\frac{c}{d}=\frac{a*c}{b*d}$$

(b) Write a program that declares two objects of class ***fraction***, and repeatedly reads data for these two objects from the keyboard, then perform the two operations on them, displays the results, and asks the user whether he/she wants to continue or not.

Question 4: **[15 Marks]**

(a) Declare and implement the following class hierarchy:

(i) An abstract base class ***Shape*** that has one data member ***Perim*** that represents the perimeter of the shape, and 3 pure virtual member functions: ***getdata()***, ***showdata()***, and ***getPerim()***.

(ii) A derived class ***Triangle*** that has 3 additional data members, representing the 3 sides of the triangle, and implements the 3 member functions: ***getdata()***, to read the 3 sides of the triangle; ***showdata()***, to display the triangle data; and ***getPerim()***, to calculate the perimeter of the triangle.

(iii) A derived class ***Rectangle*** that has additional 2 data members, representing the length and width of the rectangle, and implements the 3 member functions: ***getdata()***, to read the length and width of the rectangle; ***showdata()***, to display the rectangle data; and ***getPerim()***, to calculate the perimeter of the rectangle.

(b) Write a main program that creates an array of 10 pointers to ***Shape***. In a loop, ask the user for data about a shape and its type (triangle or rectangle), and use **new** to create a suitable object (***Triangle*** or ***Rectangle***) to hold the data, then put the pointer to the object in the array. When the user finishes entering the data for all shapes, display the data of all the shapes entered.

With My Best Wishes