INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE – 247 667

Autumn Semester 2015 – 16 **End Term Examinations**

Fundamentals of Object Oriented Programming (CSN -103) **Max. Marks: 80**

Answer ALL questions

1. (a) Write a JAVA program to check whether the given string is palindrome or not.

   (b) Write an one line code (algorithm) in JAVA to check whether given four digit year is leap or not. **(5+5)**

2. (a) Write the output of the following program? Show all the steps.

```
public class Switch2
{
    final static short x = 2;
    public static int y = 0;
    public static void main(String [] args)
    {
        for (int z=0; z < 3; z++)
        {
            switch (z)
            {
                case x: System.out.print("0 ");
                case x-1: System.out.print("1 ");
                case x-2: System.out.print("2 ");
            }
        }
    }
}
```

   (b) In an $n \times n$ θ**-matrix,** all the co-efficients other than those in row 1, column 1, row *n*, column *n* and diagonal are zero.
   (i) Give an example of 5×5 θ**-matrix**.

   (ii) How many non-zero elements can you find in an $n \times n$ θ**-matrix?**

   (iii) Write a JAVA fragment of the code to find the sum of two θ**-matrices** A and B with the help of Q 2(b)(ii). In a usual matrix addition you require n^2 steps for any $n \times n$ matrix. How many **exact steps** do you require to add two θ**-matrices**. **(6+4)**

3. (a) Write the output of the following C++ Program. Show all the steps.

```
#include <iostream>

using namespace std;

int  f(int  x,  int*  py,  int**
ppz)
 {
     int y,z;
     **ppz+=1;
     z=**ppz;
     cout<<z<<endl;
     *py+=2;
     y=*py;
     cout<<y<<endl;
     x+=3;
     cout<<x<<endl;
     return x+y+z;
  }
```

```
int main()
{
   int c, *b, **a;
   c=6;
   b=&c;
   a=&b;
   cout<<f(c,b,a)<<endl;
   return 0;
}
```

(b) Write the output of the following C++ program. Show all the steps.

```
#include <iostream>
using namespace std;
int main()
{
int arr[]={10,20,30,40,50,60,70};
int i, *ptr;
int **pptr;
for(ptr=arr+6, i=0; i<=4; i+=2)
{cout<<5*ptr[-i]+175<<endl;
pptr=&ptr;
++**pptr;
cout<<**pptr<<endl;
}
return 0;
}
```

(4+6)

4. (a) Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay Rs. 70/- toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected. Model this tollbooth with a class `tollBooth`. The two data members are a type `int` to hold the total number of cars, a type `double` to hold the total amount of money collected.

PTO

A constructor initializes both of these to 0. A method called `payingCar()` increments the car total and adds 70 to the cash total. Another method called `nopaycar(),` increments the car total but adds nothing to the cash total. Finally, a method called `display()` displays the two totals. Make appropriate methods.

Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count nonpaying car. Pushing the Esc key should cause the program to print out the total cars and total cash and then exit.

(b) Explain Method overloading with an example. **(7+3)**

5. (a) Can we declare Constructor final? Why or Why not?

   (b) Imagine a publishing company that markets both the book and CD version of its works. Create a class **publication** that stores the title (type **String**) and price (type **float**) of a publication. From this class derive two classes: **book**, which adds a page count (type **int**); and **CD**, which adds a playing time in minutes (type **float**). Each of these three classes should have a **getdata()** function to get its data from the user at the keyboard, and a **putdata()** function to display its data.
   Write a **main()** to test the **book** and **CD** classes by creating instances of them, asking the user to fill in their data with **getdata()** and then displaying the data with **putdata().**

   (c) What are abstract Classes? Differentiate between Abstract and final classes. **(2+6+2)**

6. (a) Create a JAVA package `fact_recursion` in which the corresponding class of this package should contain a recursive method that calculates a factorial of a non-negative number. Create an another package called `fact_final` and **using fully qualified name**, create an object in the corresponding class of this package. This object should invoke the recursive factorial method of a package `fact_recursion` by passing a non-negative integer.

   (b) Create two interfaces `Compute_Perimeter` and `Compute_Area` in which you have to declare abstract methods for computing perimeters of the entities: Circle, Rectangle, Square and Rhombus in `Compute_Perimeter` interface and declare abstract methods of computing areas of the above entities in `Compute_Area` interface.

   Now create a class that should use multiple inheritance to inherit both interfaces and your JAVA program should compute perimeters and areas (by creating main method in the class) of all the entities by calling the appropriate methods. **(5+5)**

7. (a) Write a JAVA program to find the rate of change of slope between ordered pairs of 2D co-ordinates. Declare a class with appropriate instance variable and methods to find the rate of change of slope = (y2-y1)/(x2-x1) [e.g. (2,3) and (5,4) is one set of ordered pairs then rate of change of slope =(4-3)/(5-2)=1/3=0.33 and second set of ordered pair is (2,3) and (4,4) then rate of change of slope = (4-3)/(4-2)=1/2=0.5. 1st set of ordered pair is having less rate of change.

   User defined exception handling required to take care for avoiding (i) rate of change of slope as infinity (ii) rate of change of slope as negative.

   (b) Write the Output of the following program

```
public class MyProgram
{
    public static void main(String args[])
    {
        try
        {
            System.out.print("Hello world ");
        }
        finally
        {
            System.out.println("Finally executing ");
        }
    }
}
```

(8+2)

8. (a) Develop an applet that generates first 200 **Isolated primes** on the screen with horizontal lines that are enclosing the output. Write a HTML page and test the applet.

   Isolated Primes: Primes p such that neither p−2 nor p+2 is prime. 2, 23, 37, 47, 53, ...are some of the examples.

   (b) Write a Java applet program that draws a Pie Chart of number of students registered in various departments at 1st year level with in IIT Roorkee. Access input that are passed through PARAM statements in HTML. (Each pie can be drawn using the `drawArc()` or the `fillArc()` method).

(5+5)