Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

A Constituent Institution of Manipal University

III SEMESTER B.TECH. (INFORMATION TECHNOLOGY / COMPUTER AND COMMUNICATION ENGINEERING)

END SEMESTER EXAMINATIONS, NOV/DEC 2016

SUBJECT: OBJECT ORIENTED PROGRAMMING [ICT 2101]

REVISED CREDIT SYSTEM
(23/11/2016)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer **ALL** the questions.
- Missing data may be suitably assumed.
- Use Java language syntax for coding.

1A. Differentiate between compile and run time polymorphism. Explain with a suitable example/examples how java supports the same. **5**

1B. Write a program to create an integer array using command line arguments and display the occurrence of two consecutive numbers in the array.
Example: command line arguments: 12 13 1 0 2 3
Expected output:
Consecutive numbers occurred at index 0 and 1 and they are 12 13
Consecutive numbers occurred at index 4 and 5 and they are 2 3 **3**

1C. What is an interface? Mention its importance. **2**

2A. Write a program that prints prime numbers between 1 and 50, with 5 numbers per line. Then the program should print "Good" in place of the prime numbers whose modulus of 3 equals 1.
The output should look like the sample given below.

```
 1      3       5      7       11
13      17      19     23      29
31     .....
Good   3       5      Good    11
Good   17      Good   23      29
Good....
```

5

2B. Write a program to perform the following operations.

i. Read a main String and two substrings passed as command line arguments.
ii. Find the number of occurrences of second substring in the main string.
iii. Replace the second substring occurrence by the first substring in the main string. **3**

2C. Write the output of the following code snippets with justification/s for the answer.

i.
```
String s1=new String("Hello");
String  s2=new String("World");
System.out.println(s1=s2);
```

ii.
```
try{
  int a=0;
  int b=5;
  int k=a/b;
  System.out.println("Hi!!");
 }
 catch(Exception e) { System.out.println("Exception");}
```

iii.
```
public static void main(String args[]) {
 try {
     statement 1;
     statement 2 ;
     statement 3 ;
     }
 catch(Exceptionl e1) { }
 catch(Exception2 e2) { }
 catch(Exception3 e3) { throw e3; }
finally
{   statement5; }
 statement4;
 }
```

If the exception is of type Exception3, what will happen? **2**

3A. Write a java program to store 100 long integers into a file named "value.dat". The program should read and display the contents. Then it should replace the long integer value positioned at multiple of 5 by number 100 if it is even or replace by 101 if it is odd and store the result in a new file.
The sample output is as follows: **5**

"Value.dat"

1234567 5674343 123456 998765 989877756 11213133 12312345 89765433 11234213 12341121 121112345.....

"New.dat"

1234567 5674343 123456 998765 100 11213133 12312345 89765433 11234213 101 121112345.....

3B. Explain the following with suitable code snippet: **3**

i. static keyword
ii. synchronized keyword

3C. What is the output for the following code snippet? Justify.

i.
```
int m=100;
while(true) {
    if(m<10)
        continue;
    m=m-10; }
System.out.println(+m);
```

ii.
```
int smallNumber = 13;
if ((smallNumber > 5) && (smallNumber < 10)){
     System.out.println("Between 5 and 10");
 }
if ((smallNumber > 10) && (smallNumber < 15)){
      System.out.println("Between 10 and 15");       }
if ((smallNumber > 15) && (smallNumber <20)) {
     System.out.println("Between 15 and 20");   }
```

iii.
```
public class Switch1 {
  final static short x = 2;
  final static int y =0;
  public static void main(String [] args) {
      for (int z=0; z<3; z++) {
          switch(z) {
              case y: System.out.print("0 ");
              case x-1: System.out.print("1 " );
              case x: System.out.print("2 ");
                 } } } }
```

2

4A. Write a java application which is used to declare the results of an examination. The application should be designed considering the classes described below.
A superclass called Person to store common attributes such as name and age. The subclasses Student and Teacher with their specific properties. For students, maintain the course names taken and their respective marks; add a course with marks, print all courses taken and the average marks. While adding the marks obtained in a course if the value is less than 0 or greater than 100 raise a user defined InvalidMarksException. A student takes no more than 4 courses. For teachers, maintain the courses taught currently, and able to add or remove a course taught. A teacher teaches not more than 2 courses concurrently.
Student class has a sub class called Result which has method to declare result and also maintains final result in the form of fail or pass. The final result is declared as pass if the average of the total marks of a student is greater than 40 else it is declared as fail.
Write a main method where you construct objects of the above designed classes and display the results. The program should not allow object creation of the superclass Person. **5**

4B. Explain the types of inheritance supported by java with relevant example/s. **3**

4C. Explain the following with suitable syntax:
i. ObjectOutputStream class
ii. Use of toString method of Object class

2

5A. Write a multithreaded program to check whether a matrix is an unit matrix (all values of the matrix are 1). Each thread should check a row of the matrix using rowcheck() method and main should display whether the matrix is unit or not after the completion of the check.
Note: The rowcheck() function called by multiple threads in the above program are independent without any dependencies(can be executed concurrently). After that their results can be used to make the final decision about the matrix. **5**

5B. Write a swing application to do the following:
i. Accepts a string
ii. Displays whether it is a palindrome or not on-click of a button.
The message can be displayed using a label and string can be accepted using textfield. **3**

5C. How does java control the visibility of variables and methods in classes and packages? **2**