Faculty of Computing

National School of Business Management

ONLINE EXAMINATION ANSWER BOOKLET

DECLARATION:

I acknowledge the NSBM Examination Honour Code, and **I hereby confirm** that the submitted work is entirely my own and I have not

(i) used the services of any agency or person(s) providing a specimen, model, or ghost-written work in the preparation of the work I submit for this open-book examination.

(ii) given assistance in accessing this paper or in providing a specimen, model or ghost-written work to other students submitting for this open-book examination.

□ I have marked the attendance

INDEX NO	19747	
NAME	D.G.L.S.Daluwathumulla	
DEGREE PROGRAM	BSc in Management Information Systems (Special)	**BATCH** 20.1
MODULE CODE	CS101.3	
MODULE	Object oriented programming	

INSTRUCTIONS:

Use this answer booklet to answer structured essay type questions only.

When you upload to the LMS, you need to convert this file into PDF format and save with your index no.

All the submissions will be checked for plagiarism. Plagiarism, collusion, and copying are severe and grave offences in the university and penalties that would be imposed.

01).

1.

```
public class Employee{
private String name;
 private int perHourRate;
private int perDayRate;


 public String getEmpName(){
 return name;
 }
 public int setPerHourRate(int hourRate) {
             this.perHourRate=hourRate;
 return perHourRate;
 }
public int getPerHourRate() {
 return perHourRate;
 }
 public int getPerDayRate(){
 perDayRate=perHourRate*24;
 return perDayRate;
 }


}
```

2.

```
public class Employee{
private String name;
```

```
 private int perHourRate;
private int perDayRate;


 public String getEmpName(){
 return name;
 }
 public int setPerHourRate(int hourRate) {
            this.perHourRate=hourRate;
 return perHourRate;
 }
public int getPerHourRate() {
 return perHourRate;
 }
 public int getPerdayRate(){
perDayRate=perHourRate*24;
 return PerDayRate;
 }
  //default constructor
 public Employee()
 {
System.out.println("default")
 }
  //parameterized constructor
 public Employee(String  empname)
 {
            this.name=empname;
 }


}
```

3.

```
public class EmployeeDemo {


        public static void main(String[] args){
          Employee objwithDefCons = new Employee();
        Employee objwithParaCons = new Employee("Nimal");


        }


}
```

4.

```
public class EmployeeDemo {


        public static void main(String[] args){
          Employee objwithDefCons = new Employee();
        Employee objwithParaCons = new Employee("Nimal");


        setPerHourRate(3);
        System.out.println("Employee Name: " + objwithParaCons .getEmpName());
        System.out.println("Per Hour Rate: " + objwithParaCons .getPerHourRate());
        System.out.println("Per Day Rate: " + objwithParaCons .getPerDayRate());
        }
}
```

02).

1.

```
public class A{
  int a;
  //parameterized constructor
 public A(int  aIn)
 {
            this.a=aIn;
  }
}

public class B extends A{
int b,c;
int resultMulty;
public int multiplyMeth(){
resultMulty=a*b*c;
System.out.println( resultMulty);
return resultMulty;
}
 public B(int aIn,int  bIn,int cIn)
 {

            Super(a)=aIn;
            this.b=bIn;
```

```
        this.c=cIn;
  }

}
```

2.

```
public class C {

        public static void main(String[] args){
          B obj = new B(4,5,6);

          System.out.println("Multiplication of a,b,c: " + obj.multiplyMeth());

        }

}
```

3.

```
abstract class Employee{
String dept;
public abstract void payAllowance(){};

 public void displayDepartment() {

 System.out.println (dept);
 }
}
```

```
interface Pay{
	void payBonus(){};
}

class EmployeePay implements Pay extends Employee{
	@Override
	public  void payAllowance(){
	System.out.println("payAllowance");//sample statements
	};
	@Override
	public  void payBonus(){
	System.out.println("payBonus");	//sample statements
	};

	public static void main(String args[]){
	EmployeePay obj = new EmployeePay();
	obj.displayDepartment();
	obj.payAllowance();
	obj.payBonus();

 }
}
```

03).

1.

```
import java.util.Arrays;
import java.util.Scanner;

public class MultiExcCatcher {
  public static void main(String [] args) {
    Scanner sc = new Scanner(System.in);
    int[] arrCheckExc = {10, 20, 30, 2, 0, 8};
    System.out.println("Enter 3 integers ");
    System.out.println("Array: "+Arrays.toString(arr));
    System.out.println("pls select a numerator and denominator (not 0) from this array,
    (enter positions 0 to 5)");
    int a = sc.nextInt();
    int b = sc.nextInt();
    try {
      int resultDivision = (arrCheckExc[a])/(arrCheckExc[b]);
      System.out.println("Result of "+arrCheckExc [a]+"/"+arrCheckExc [b]+": "+resultDivision );
    }
    catch(ArrayIndexOutOfBoundsException | ArithmeticException exp) {
      System.out.println("Warning: Enter inputs as guided .Be careful about the number types ");
    }
  }
}
```

2.

Class mythread

```
public class MyThread

{



   public static void main(String[] args)

   {

       MyClassA m1=new MyClassA();

       m1.start();





       MyClassB m2=new MyClassB();
```

```
        M2.start();

    }

}
```

Class mhyclass

```
public class MyClassA extends Thread
{
        public void run()
        {
          int x;

          for(x=1;x<=5;x++){
              System.out.println("Thread A  "+ x);
}
        }
}
public class MyClassB extends Thread
{
        public void run()
        {
          int y;
```

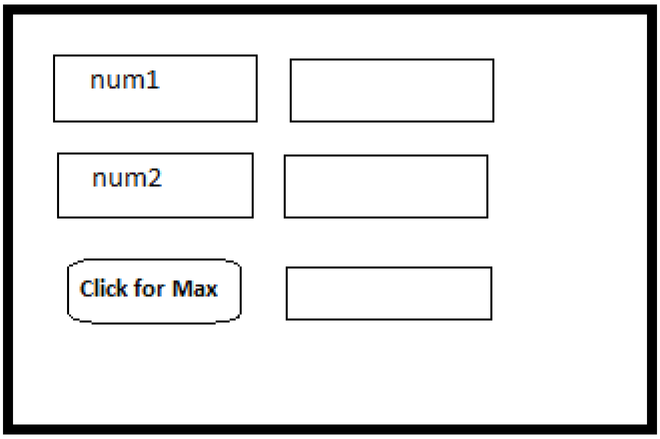
```
     for(y=1;y<=5;y++)


         System.out.println("Thread B  "+ y);

    }

}
```

04).



1.
Two text areas for num1,num2
One button
Display label-Maxlbl

```
float  num1,num2;
private void ClickbtnActionPerformed(java.awt.event.ActionEvent evt) {
maxNo= Math.max(num1,num2);
Maxlbl.setText(" "+maxNo);
}
```

2.

```
import java.io.FileWriter;
import java.io.IOException;

public class WriteToFile {
```

```
public static void main(String[] args) {
try {
FileWriter myWriter = new FileWriter("OOP.txt");
myWriter.write("Object-Oriented Programming with Java");
myWriter.close();
} catch (IOException e) {
System.out.println("An error occurred.");
e.printStackTrace();
}
}
}
```

3.

```
String queryStr="SELECT * FROM Employee";
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(queryStr);

Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select Employee_ID,First_Name,Last_Name,Basic_salary,Dept_ID
from Employees");

while(rs.next()){
System.out.println(rs.getInt(“Employee_ID"));
System.out.print(rs.getString(2)+":");
System.out.print(rs.getString(3)+":");
System.out.println(rs.getDouble(“Basic_salary"));
System.out.println(rs.getInt(“Dept_ID"));
}
```