B.M.S COLLEGEOF ENGINEERING, BANGALORE-19
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

INTERNALS -II

Course Code: 21CS3PCOOJ	**Course Title:** Object Oriented Java Programming	
Semester: III	**Maximum Marks: 40**	**Date:20/1/2023**

PART-A

No.	Question	Marks
1.	Demonstrate the usage of keywords throw and throws in Exception handling using an example program. Explanation of keyword throw and throws in exception handling with example	5 5 marks

PART-B

No.	Question	Marks
2.a	Analyze the given program find the errors. Write the corrected program. Underline the places where errors are corrected.	5

```
interface MyInterface
{
  void method1();
  void method2();
}
class Demo extends MyInterface
{
  void method1()
  {
       System.out.println("implementation of method1");
  }
  void method2()
  {
       System.out.println("implementation of method2");
  }
  public static void main (String arg[])
  {
       MyInterface obj = new MyInterface ();
       obj. method1();
  }
}
```

```
interface MyInterface{
 void method1();
  void method2();
}
Class Demo implements MyInterface{
  public void method1(){ S.o.p("-----");} public void method2(){ S.o.p("-----");
    psvm(String args[]){ MyInterface obj;  obj=new Demo(); obj1.method1();
      obj.method2();}}
```

2mark + 3 mark

Underlining shows where the error is 2 mark and adding extra code to rectify output 3mark

2.b	Analyse the program and determine the output when the variable a1=0 and a1=1. Justify your answer with appropriate reason.
	
	`class Exceptionhan {`
	`public static void main (String args[]){`
	`int a1=1;`
	`try {`
	`try {`
	`System.out.println("going to divide");`
	`int b =39/a1;`
	
	`try {`
	`int a[]=new int[5];`
	`a[5]=4;`
	`} catch (ArithmeticException e) {System.out.println(e);}`
	
	`} catch (ArrayIndexOutOfBoundsException e) {System.out.println("Array overflow");}`
	
	`System.out.println("other statement");`
	`} catch (Exception e) {System.out.println("handeled");}`
	
	`System.out.println("normal flow..");`
	`}`
	`}`
	
	Expected Output: when a1=1 Going to Divide,Array Overflow ,other statement and then Prints Normal Flow.
	In the Innermost try error is generated search for immediate catch if not available search through stack frames and then prints it .Same reason for a1=0 is taken.
	When a1=0 prints Going to divide Handled and prints Normal flow.
		5
		
		2.5+ 2.5 marks

2.c	*Modify the given program to get the output shown.*
	Program:
	`class A extends Thread { …… }`
	`class Main{`
	`public static void main(String args[]){`
	`A a1=new A();`
	`System.out.println("Name of thread 't':"+ t.getName( ));`
	`}`
	`}`
	
	Expected Output:
	Name of thread 't': FirstThread
	New name of thread 't': NewThread
	Thread is running.
	
	Modified Code
	`class A extends Thread{`
	`A()`
	`{`
	`super("FirstThread");`
	`start();`
	`}`
	`public void run()`
	`{`
	`System.out.println("Name of thread 't':"+getName());}`
	`public static void main(String args[]){`
	`A obj=new A();`
	`Thread t =Thread. currentThread();`
	`t.setName("NewThread");`
	`try{`
	`Thread.sleep(1000);`
	`}`
	`catch(InterruptedException e)`
	`{`
	`System.out.println("Caught");`
	`}`
	`System.out.println("New Name of thread 't':"+ t.getName());`
	`System.out.println("Thread is Running");`
	`} }`
		5
		
		2 marks
		
		2 marks
		
		1 mark

PART- C

No.	Question -	Mark

3.a.	Develop a Java Program to create three Threads using Implementable runnable Interface. Make all threads to execute for five iterations. Set the name of the three threads as "FIRST", "SECOND", "THIRD". Make the second thread to terminate for the 4th Iteration and last thread to sleep for two seconds after two iterations.	**10**

Create multiple threads.

```
class NewThread implements Runnable {

String name; // name of thread

Thread t;
NewThread(String threadname) {

name = threadname;
t = new Thread(this, name);
System.out.println("New thread: " + t);
t.start(); // Start the thread
}
// This is the entry point for thread.
public void run() {
  String s =t.getName();
if (s.equals("Two")){
for(int i = 5; i > 0; i--) {
  if(i==4) {
   t.stop();
System.out.println(name + ": " + i);
}
//Thread.sleep(1000);
}

if (s.equals("Three")){
for(int i = 5; i > 0; i--) {
System.out.println(name + ": " + i);
  if(i==3) {
try{
Thread.sleep(2000);
}// end of try
catch (InterruptedException e) {
System.out.println(name + "Interrupted");
}
}
//System.out.println(name + " exiting."); }}}}
class MultiThreadDemopri {
public static void main(String args[]) {
new NewThread("One"); // start threads
new NewThread("Two");
new NewThread("Three");
try {
// wait for other threads to end
Thread.sleep(10000);
} catch (InterruptedException e) {
System.out.println("Main thread Interrupted");
}
System.out.println("Main thread exiting.");
}}
```

Or Any other Logic of program Students Write can take into consideration.

2 mark

4 mark

4 mark

2 marks for NewThread+4 marks for implementing Run() method +4 marks for main program.

3.b.	Implement two Stacks using array of integers of size six each. Include push( ), pop( ) and display methods using the concept of Interface. Check the appropriate conditions for Stack overflow and Stack Underflow. The other condition is if Top1 and Top2 of the two stacks is equal to max of 10 elements then perform pop( ) from these two stacks and store it in a new array. Check for all combinations. Declaring Interface with methods Push ,pop() and display(). Implementing them In the class creating two stacks of length 6 in class Constructor. Implements Push1 method in class chk for overflow for stack 1, Implements Push2 method in class chk for overflow for stack 2 Implements pop method if top1 of stck 1and top2 of stck2 is equal to 10 then pop from the stack. Appropriate Logic of the program should be given through methods even if the Conditions are not mentioned in full.	**10** **1** **2+2+2** **3** **marks**

4.a.	Create three threads using any one of the two techniques. Set the priority of Threads to 10, 5 and 1 for the three threads respectively. The thread whose priority is 10, computes sum of numbers - one to five, the thread which has minimum priority computes product of numbers - one to five. Make the second thread with priority 5 to sleep for two seconds.	**10**

```
class A extends Thread
 {  int sum;
     A(){sum=0;}
     public void run()
     {
        System.out.println("Thread A started");
       for(int i=1;i<=5; i++)
       {
           sum=sum+i;}
       System.out.println("\t From  Thread A: sum= "+sum);
         System.out.println("Exit from A");
      }
   }
     class C extends Thread
    {
        int prod;
         C() {prod=1;}
        public void run()
        {
         System.out.println("Thread C started");
           for(int j=1;j<=5;j++)
           {
              prod=prod*j; }
          System.out.println("From Thread C:product="+prod);
          System.out.println("Exit from C"); }}

     class B extends Thread
     {
       public void run()
       {
            System.out.println("Thread B started");
             for(int k=1;k<=4;k++)
             {
                System.out.println("\t From thread B : K="+k);
             }
             System.out.println("Exit from B");}}

      class ThreadPrioritytest2{
            public static void main(String args[])
           {
               A tha=new A();
               B thb=new B();
               C thc=new C();
              tha.setPriority(Thread.MAX_PRIORITY);
              thc.setPriority(Thread.MIN_PRIORITY);
              thb.setPriority(Thread.NORM_PRIORITY );
              System.out.println("Start Thread A");
              tha.start();
              System.out.println("Start Thread B");
              thb.start();
              System.out.println("Start Thread C");
               thc.start();
             System.out.println("End of main Thread");}}
```

2+2+2 marks

4 mark

Or Any other Logic of the program Students Write can take into consideration.

2 marks for each Thread of class A,B and Class c using Thread class
+4 marks for main program

4.b.	Write a Java program that demonstrates user defined exception handling in inheritance tree. For example, create a base class called "Vehicle" and derived class called "MyCar" which extends the base class. In Vehicle Class, implement a constructor which takes the model, make and vehicleyear as parameters and throws the exception BadVehicle( ) when the input vehicleyear<2002. MyCar class has an instance variable engineyear, implement a constructor that throws an exception if MyCar's engineyear >=vehicleyear.	**10**

```
Class BadVehicle Extends Exception{
   String det;
    Badvehichle(String details)
  {    det =details; }

   Public String toString()
 {    System.out.println("Bad Vehicle Selected"+det);
      }
}

Class Vehicle{
 String Model,make; int vehYear;
  Vehicle(String Mo,String make,int Year){
    Model=Mo; this.make=make; vehYear=year;}

      method1() throws BadVehicle {
       if (vehYear<2002 )
        throw BadVehicle("Selection");
        System.out.println("End of method1");
}
Class MyCar extends Vehicle{
      int engineyear;
       MyCar(String S,String G,int year,int engineyear)
       {
          super(S,G,year);
           this.engineyear=engineyear;
          }
               method2() throws BadVehicle{
                     try{
                        Super.method1();
                   }
   Catch(BadVehicle e)
                    {   System.out.println("Caught"+e);}
                      If (engineyear>super.vehicleyear){
                         Throw BadVehicle("Selection");
                        }
                    }
Class userexcption{
            psvm(String args[]){
             MyCar m1=new MyCar("Swift","Maruti",2016,2020);
            // Mycar m2=new MyCar("Kwid","Renault",2018,2012);
             try{
                    m1.method2();  }
                     }
   Catch(BadVehicle e)
     {  System.out.println("Caught"+e);}
 System.out.println("End of main");}}
```

2 mark

3 mark

3 mark

2 mark

Or Any other Logic of the program Student Write can take into consideration.