ASSIGNMENT-LAB 07

Course Code: CSE-2340 **Course Title:** Software Development 1
Course Teacher: Md. Mahadi Hassan

Name: Ezaz Ahmed **ID:** C223009 **Section:** 3AM

Problem 01: Try-catch-finally.

Answer:

```
import java.util.*;

public class tc
{
    public static void main(String[] args)
    {
        try
        {
            int result = 1 / 0;
        }
        catch (ArithmeticException a)
        {
            System.out.println("An exception occurred: " +
a.getMessage());
        }
        finally
        {
            System.out.println("Finally block executed.");
        }
    }
}
```

Problem 02: Create a new exception.

Answer:

```
import java.util.*;
class CE extends Exception
{
    public CE(String message)
    {
        super(message);
    }
}

public class C
{
    public static void main(String[] args)
    {
        try
```

```
        {
            throw new CE("This is a custom exception.");
        }
        catch (CE a)
        {
            System.out.println("Found: " + a.getMessage());
        }
    }
}
```

Problem 03: Thread using Thread class.

Answer:

```
import java.util.*;

class th extends Thread
{
    public void run()
    {
        System.out.println("Thread is running.");
    }
}

public class TH
{
    public static void main(String[] args)
    {
        th k = new th();
        k.start();
    }
}
```

Problem 04: Thread using Runnable interface.

Answer:

```
import java.util.*;

class R implements Runnable
{
    public void run()
    {
        System.out.println("Thread is running.");
    }
}

public class TUR
{
    public static void main(String[] args)
    {
        Thread t = new Thread(new R());
```

```
        t.start();
    }
}
```

Problem 05: **Thread sleep method.**

Answer:

```
import java.util.*;
public class SL
{
    public static void main(String[] args)
    {
        System.out.println("A");
        try
        {
            Thread.sleep(5000);
        }
        catch (InterruptedException a)
        {
            a.printStackTrace();
        }
        System.out.println("B");
    }
}
```

Problem 06: **Thread priority.**

Answer:

```
import java.util.*;
class tp extends Thread
{
    public tp(String name)
    {
        super(name);
    }

    public void run()
    {
        for (int i = 1; i <= 5; i++) {
            System.out.println(getName() + " is running with
priority " + getPriority());
        }
    }
}

public class TP
{
    public static void main(String[] args)
    {
        tp t1 = new tp("Thread 1");
```

```
        tp t2 = new tp("Thread 2");

        t1.setPriority(Thread.MIN_PRIORITY);
        t2.setPriority(Thread.MAX_PRIORITY);

        t1.start();
        t2.start();
    }
}
```

Problem 07: **Suspend-resume.**

Answer:

```
import java.util.*;
class t extends Thread
{
    public void run()
    {
        for (int i = 1; i <= 5; i++)
        {
            System.out.println("Thread is running: " + i);
            try
            {
                Thread.sleep(1000);
            }
            catch (InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    }
}

public class SR
{
    public static void main(String[] args)
    {
        t a = new t();
        a.start();

        try
        {
            Thread.sleep(4000);
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }
```

```
        a.suspend();
        System.out.println("Thread suspended for 5 seconds.");

        try
        {
            Thread.sleep(5000);
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }

        a.resume();
    }
}
```