

## Unit 4 code

- **Exception Handling: try, catch, throw, throws, finally**
- **Built-in Exceptions**
- **Custom Exceptions**
- **Multithreading using Thread class and Runnable interface**
- **Thread Lifecycle**
- **Thread Priorities**
- **Synchronization**

### ☒ 1. Exception Handling (try, catch, throw, throws, finally)

```
java
CopyEdit
public class TryCatchExample {
    public static void main(String[] args) {
        try {
            int result = 10 / 0; // ArithmeticException
        } catch (ArithmeticException e) {
            System.out.println("Can't divide by zero!");
        } finally {
            System.out.println("This will always run");
        }
    }
}
```

---

### ☒ 2. Built-in Exceptions

```
java
CopyEdit
public class BuiltInExceptions {
    public static void main(String[] args) {
        String str = null;

        try {
            System.out.println(str.length()); // NullPointerException
        } catch (NullPointerException e) {
            System.out.println("Null object can't be used!");
        }
    }
}
```

---

### ☒ 3. Custom Exception

```
java
```

```

CopyEdit
class AgeException extends Exception {
    AgeException(String message) {
        super(message); // Pass message to parent
    }
}

public class CustomExceptionExample {
    static void checkAge(int age) throws AgeException {
        if (age < 18) {
            throw new AgeException("You are under 18!");
        } else {
            System.out.println("Eligible to vote");
        }
    }

    public static void main(String[] args) {
        try {
            checkAge(16);
        } catch (AgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

---

## ☒ 4. Multithreading using Thread class

```

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

public class ThreadClassExample {
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        t1.start(); // starts thread
    }
}

```

---

## ☒ 5. Multithreading using Runnable interface

```

java
CopyEdit
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("Runnable thread is running");
    }
}

public class RunnableExample {
    public static void main(String[] args) {
        Thread t = new Thread(new MyRunnable());
        t.start();
    }
}

```

```
}
```

---

## ☒ 6. Thread Lifecycle (States)

```
java
CopyEdit
class ThreadLife extends Thread {
    public void run() {
        System.out.println("Running...");
    }
}

public class ThreadLifecycle {
    public static void main(String[] args) {
        ThreadLife t = new ThreadLife();
        System.out.println("State before start: " + t.getState());
        t.start();
        System.out.println("State after start: " + t.getState());
    }
}
```

---

## ☒ 7. Thread Priorities

```
java
CopyEdit
class PriorityThread extends Thread {
    public void run() {
        System.out.println("Thread running with priority: " +
getPriority());
    }
}

public class ThreadPriorityExample {
    public static void main(String[] args) {
        PriorityThread t1 = new PriorityThread();
        t1.setPriority(Thread.MAX_PRIORITY); // 10
        t1.start();
    }
}
```

---

## ☒ 8. Synchronization

```
java
CopyEdit
class Counter {
    int count = 0;

    synchronized void increment() {
        count++;
    }
}

public class SyncExample {
    public static void main(String[] args) throws InterruptedException {
```

```
Counter c = new Counter();

Thread t1 = new Thread(() -> {
    for (int i = 0; i < 1000; i++) c.increment();
});

Thread t2 = new Thread(() -> {
    for (int i = 0; i < 1000; i++) c.increment();
});

t1.start();
t2.start();
t1.join();
t2.join();

System.out.println("Final Count: " + c.count);
}
}
```