Q1. Explain Exception Handling in java Programming?

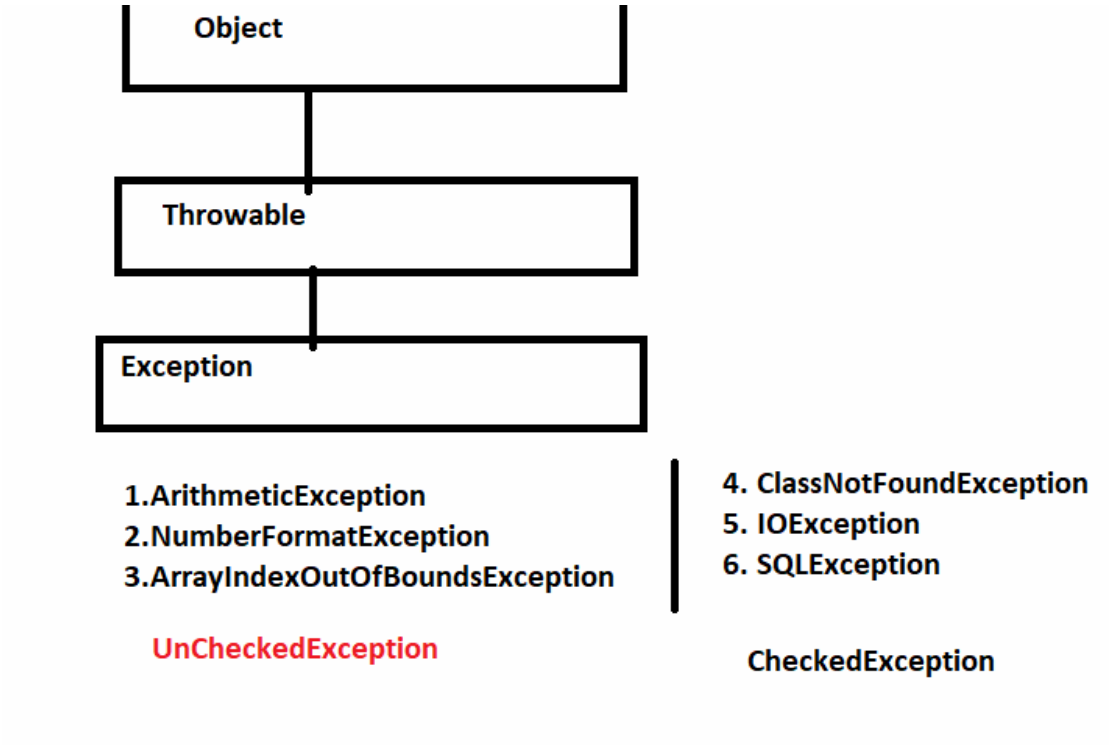
Ans: Exception Handling in java is a mechanism to handle run time errors(exception) in order maintain normal flow of the program.It helps in managing exceptions(unexpected event) and avoiding termination of the program during execution

Throwable: The super class for all exceptions and error.

Exception: It is a child class of Throwable Represents a conditions that a program might want to catch.

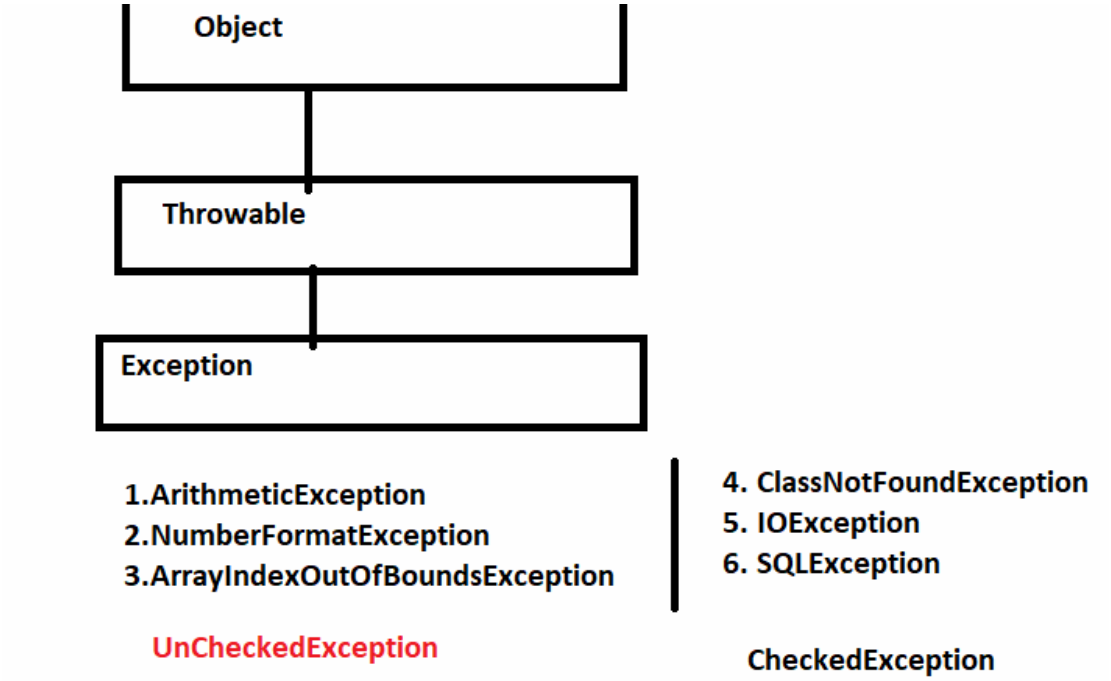
There are two types Exception
1. Checked Exception : Handle to Mandatory

2. Unchecked Exception: Handle to Optional



There are two types of Exceptions
1. Pre-Defined Exception
2. User Defined Exception

Errors: Represents serious problem that application should not attempt to catch(outOfMemoryError, stackOverFlow)

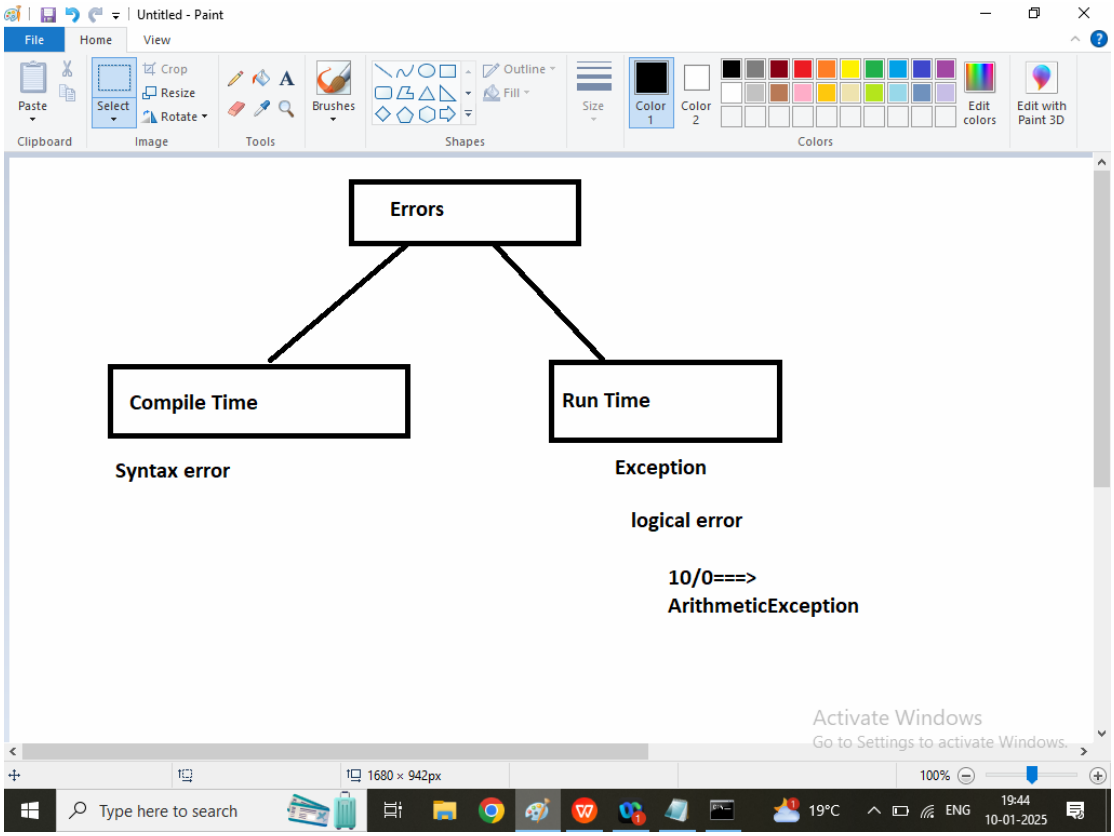


```
class E1{
	public  static void main(String args[]){
	System.out.println("Hello...hi");
	int a,b,c;
	a=10;
	b=0;
	System.out.println("before Exception");
	c=a/b;
	System.out.println(c);
	System.out.println("Hello...Bye");
	System.out.println("After Exception");

	}
}
```

```
class E1{
	public  static void main(String args[]){
	System.out.println("Hello...hi");
	int a,b,c;
	a=10;
	b=0;
	System.out.println("before Exception");
	c=a/b;
	System.out.println(c);
	System.out.println("Hello...Bye");
	System.out.println("After Exception");

	}
}
```



There are 5 keywords to handle exception in java

1. try:
2. catch:
3. finally
4. throw
5. throws

Syntax: (try and catch)

```
try{
//code that result may be exception

}catch(Exception or Its Derived Class){
//handling code

}
```

Syntax: (try..finally)

```
try{
//code
}finally{

}
```

```
try{
//code
}catch(){
//handling codetry{
//code
}catch(){
//
}

finally{

}
```

Q1. Explain Exception Handling Keywords in java?

1. try:defines the block of code to detection of exception
2. catch: defines a block of code to handle specific exception
3. finally: defines block of code that always will be executed. finally is unconditional, we can use only one finally after try
4. throw: used to explicitly throw an exception
5. throws: declares exceptions that a method can throw to the caller.

```
class E1{
	public  static void main(String args[]){
	System.out.println("Hello...hi");
	int a,b,c=0;
	a=10;
	b=2;
	System.out.println("before Exception");
```

```
	try{
	System.out.println("Enter Try");
	c=a/b;

	System.out.println("Exit try");
	}catch(ArithmeticException ae){
	System.out.println("This is catch
block");
	System.out.println("Denominator
should not be zero");
	}
	System.out.println(c);
	System.out.println("Hello...Bye");
	System.out.println("After Exception");

	}
 }
```

```
class E1{
	public  static void main(String args[]){
	System.out.println("Hello...hi");
	int a,b,c=0;
	a=10;
	b=0;
```

```
        System.out.println("before
Exception");
        try{
        System.out.println("Enter Try");
        c=a/b;

        System.out.println("Exit try");
        }catch(ArithmeticException ae){
        System.out.println("This is catch
block");
        System.out.println("Denominator
should not be zero");
        }
        System.out.println(c);
        System.out.println("Hello...Bye");
        System.out.println("After Exception");

        }
 }
```

```
class E1{
        public  static void main(String args[]){
        System.out.println("Hello...hi");
        int a,b,c=0;
```

```
        a=10;
        b=2;
        System.out.println("before
Exception");
        try{
        System.out.println("Enter Try");
        c=a/b;

        System.out.println("Exit try");
        }finally{
        System.out.println("This is finally
block");

        System.out.println(c);
        System.out.println("Hello...Bye");
        System.out.println("After Exception");
        }
        }
 }
```

```
class E1{
        public  static void main(String args[]){
        System.out.println("Hello...hi");
        int a,b,c=0;
```

```
        a=10;
        b=0;
        System.out.println("before
Exception");
        try{
        System.out.println("Enter Try");
        c=a/b;

        System.out.println("Exit try");
        }finally{
        System.out.println("This is finally
block");

        System.out.println(c);
        System.out.println("Hello...Bye");
        System.out.println("After Exception");
        }
        }
 }
```

```
class E1{
        public  static void main(String args[]){
        System.out.println("Hello...hi");
        int a,b,c=0;
```

```
        a=10;
        b=0;
        System.out.println("before
Exception");
        try{
        System.out.println("Enter Try");
        c=a/b;

        System.out.println("Exit try");
        }finally{
        System.out.println("This is finally
block");

        System.out.println(c);
        System.out.println("Hello...Bye");
        System.out.println("After Exception");
        }
        }
 }
```