ASSIGNMENT-6

COURSE CODE-CS261

COURSE NAME-OBJECT ORIENTED DESIGN AND PROGRAMMING

1. Predict the output:

```
public class TryCatchExample1{
public static void main(String args[]){
try{
try{
System.out.println("going to divide");
int b =39/0;
}catch(ArithmeticException e){System.out.println(e);}
try{
int a[]=new int[5];
a[5]=4;
}catch(ArrayIndexOutOfBoundsException
e){System.out.println(e);}
System.out.println("other statement");
}catch(Exception e){System.out.println("handeled");}
System.out.println("normal flow..");
}
}
```

This is called a nested try block.

ANS. The output of the above program would be:-

```
going to divide
java.lang.ArithmeticException: / by zero
java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
other statement
normal flow..
```

2. Demonstrate the difference between finally block and catch block.

ANS. Catch block is a block which would only be executed when the exception is thrown by try block .Finally block is a block which would always be executed whether the exception is thrown by try block or not.

3. Write a java program to create a validate method that takes integer value as a parameter. If the age is less than 18, throw the ArithmeticException otherwise print a message welcome to vote.

ANS.

```
import java.util.*;
class A
{
	public void valid(int n)
	{
		try
		{
			if(n<18)
			{
				throw new ArithmeticException("age is less than 18");
			}
			System.out.println("welcome to vote");
		}
		catch(ArithmeticException e)
		{
			System.out.println(e);
		}
	}
}
class Q3
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter the age");
		int n;
```

```
                n=sc.nextInt();
                A ob=new A();
                ob.valid(n);
        }
}
```

4. Predict the output:

```
public class TryCatchExample1{
void m(){
int data=50/0;
}
void n(){
m();
}
void p(){
try{
n();
}catch(Exception e){System.out.println("exception
handled");}
}
public static void main(String args[]){
TryCatchExample1 obj=new TryCatchExample1();
obj.p();
System.out.println("normal flow...");
}
}
```

This is called Exception Propagation.

ANS. The output of the above program would be:-

```
exception handled
normal flow...
```

5. Write a java program to check whether the checked exceptions are propagated not.
ANS.

```
import java.io.IOException;
class Simple {
void m() throws IOException
{
throw new IOException("device error");
}
void n() throws IOException
{
m();
}void p()
{
try {
n();
}
catch (Exception e) {
System.out.println("exception handled");
}
}
public static void main(String args[])
{
Simple obj = new Simple();
obj.p();
System.out.println("normal flow...");
}
}
```

6. Predict the output:

```
import java.io.IOException;
class Testthrows1{
void m()throws IOException{throw new IOException("device
error");//checked exception
}
void n()throws IOException{
m();
}
void p(){
try{
n();
}catch(Exception e){System.out.println("exception handled");}
}
public static void main(String args[]){
Testthrows1 obj=new Testthrows1();
obj.p();
System.out.println("normal flow...");
}
}
```

ANS. The output of the above program would be:-

```
exception handled
normal flow...
```

7. What is the issue with the below java code?

```
import java.io.*;
class M{
void method()throws IOException{
throw new IOException("device error");
}
}
class Testthrows4{
public static void main(String args[])throws
IOException{//declare
exception
M m=new M();
m.method();
System.out.println("normal flow...");
}
}
```

ANS. The issue with the above java code is that we are not handling the exception which is thrown by ‘Class M’.The exception can be handled by using the “try and catch block” from where the method called.So ,the above code will produce run time error which is:-

```
Exception in thread "main" java.io.IOException: device error
	at M.method(Testthrows4.java:4)
	at Testthrows4.main(Testthrows4.java:10)
```

NAME-AMAN KUMAR KANOJIA
ROLL NO.-201851014
SECTION-1
GROUP-1A