NAUT B05 - Java Assessment

Name: REUBEN THOMAS

Ps No: 10685230

Date: 24/01/2022

Assessment

Java Arrays

1. Declare an Integer type Array with size 5 and initialize with integer values:
    a. Reverse the Array.
    b. Perform Addition, Subtraction, and Multiplication operations on original array and reversed array.

        i.e., let's say given array is: [4,5,6,7,8]

        reverse array would be: [8,7,6,5,4]

        now we have two values a = 45678 and b = 87654

        so, addition would be a+b= 45678+87654 = 133332

Java String

1. Write a java program where you need to take a String Input from a user and after that you need to fetch all the Digit characters from the string and make the largest number possible for those digits. If there is only one digit, then you just need to return that digit and if no digit available then return 0.

    i.e., input String is "ab2gh4ut7iuy"

    then digit chars would be 2,4 and 7

    and the largest no would form from these digit is : 742

Java Regex

1. Using Java regex Validate the pattern given below:
    a. 10 Digits Phone no.
    b. A password which should at least have one upper case letter, lower case letter, Special Character, and digit. The length of the password length should not be less than 7 and not more than 15.

Java Exceptional Handling

2. Write a java program where you need to use a:
    a. Multiple catch blocks with try block.
    b. Finally Block.
    c. Create your own custom exception.

OOPS

3. Write a Java programs where you need to implement
    a. Multilevel Inheritance
    b. Hierarchical Inheritance
    c. This keyword.

    Note: use exceptional handlings wherever there is a possibility of Exception.

Note: Share all your code and output screenshot.

Solutions:

Java Array

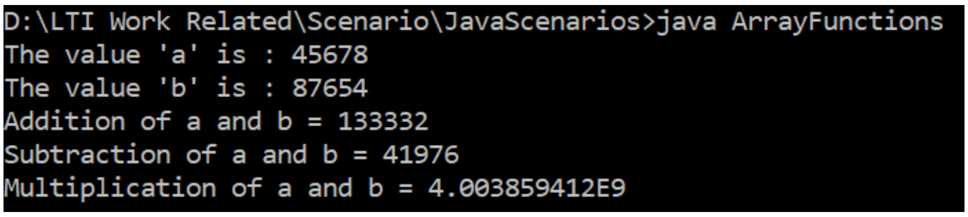
1.

Code:

```
public class ArrayFunctions
{
   public static void main(String[] args)
{
       int[] a={4,5,6,7,8};
       int[] b=new int[5];
       int i,atot=0,btot=0,shiftby,sum=0,diff=0;
       int j=a.length-1;
       double multi=0;
       for(i=0;i<a.length;i++)
            {
          b[j]=a[i];
            j=j-1;
             }

       for(i=0;i<a.length;i++)
{
          shiftby=(int) Math.pow(10,i);
          atot+=b[i]*shiftby;
          btot+=a[i]*shiftby;
if (i==a.length-1)
{
System.out.println("The value 'a' is : "+atot);
System.out.println("The value 'b' is : "+btot);
}
       }
```

```
        sum = atot+btot;

diff = btot-atot;

multi = (double)atot*btot;

        System.out.println("Addition of a and b = "+sum);

System.out.println("Subtraction of a and b = "+diff);

System.out.println("Multiplication of a and b = "+multi);

    }

}
```

Output:



```
D:\LTI Work Related\Scenario\JavaScenarios>java ArrayFunctions
The value 'a' is : 45678
The value 'b' is : 87654
Addition of a and b = 133332
Subtraction of a and b = 41976
Multiplication of a and b = 4.003859412E9
```

Java Strings:

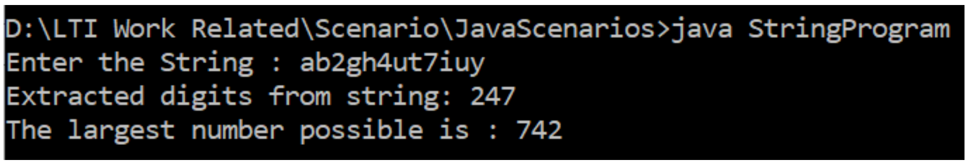
1.

Code:

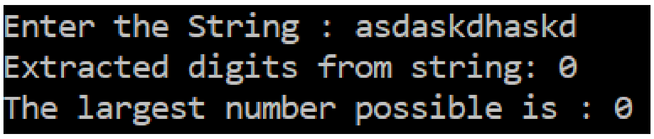
```
import java.util.*;

import java.util.regex.*;

public class StringProgram{

static int findMaximum(int arr[], int n)

  {

      Arrays.sort(arr);

      int num = 0;

      for(int i = n - 1; i >= 0; i--)

      {

         num = num * 10 + arr[i];

      }

      return num;

  }



  public static String getDigits(String strValue)

{

      String str = strValue.trim();

      String digit="";

      for (int i = 0; i < str.length(); i++) {

         char c = str.charAt(i);

         if (Character.isDigit(c))

            digit = digit+c;

      }

if (digit=="")

{

return "0";

}
```

```
else{
return digit;
}


  }



  public static void main(String args[])
  {
     Scanner sc= new Scanner(System.in);
     System.out.print("Enter the String : ");
     String str= sc.nextLine();
     sc.close();

     String d=getDigits(str);
     System.out.println("Extracted digits from string: "+d);

     int[] f= new int[d.length()];
     String[] df = d.split("");
     for(int i=0;i< df.length;i++)
{
        f[i]=Integer.parseInt(df[i]);
     }
     int n = f.length;
if (n>1)
{
     int largest = findMaximum(f, n);
     System.out.println("The largest number possible is : "+largest);
}
else{
System.out.println("The largest number possible is : "+f[0]);
}
   }


}
```

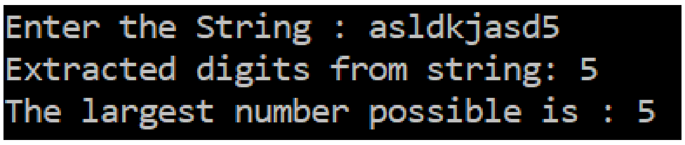
Output:



```
D:\LTI Work Related\Scenario\JavaScenarios>java StringProgram
Enter the String : ab2gh4ut7iuy
Extracted digits from string: 247
The largest number possible is : 742
```



```
Enter the String : asdaskdhaskd
Extracted digits from string: 0
The largest number possible is : 0
```



```
Enter the String : asldkjasd5
Extracted digits from string: 5
The largest number possible is : 5
```

Java Regex

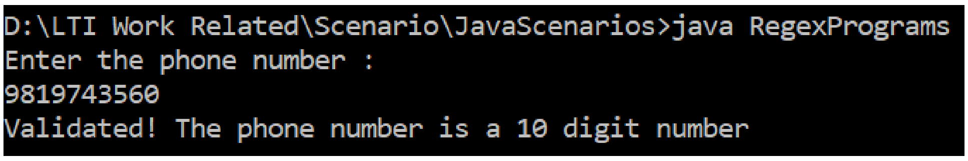
1.

a. Phone number digits

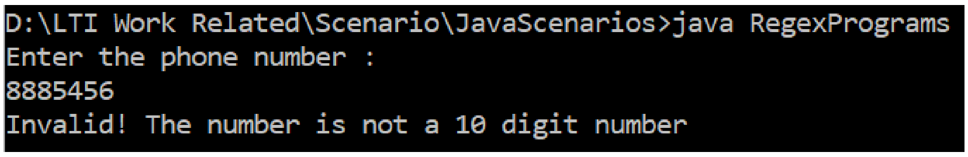
Code:

```
import java.util.*;

import java.util.regex.*;

public class RegexPrograms{


public static void main(String[] args)

{

Scanner sc= new Scanner(System.in);

System.out.println("Enter the phone number : ");

String phno= sc.nextLine();

sc.close();

Pattern pattern = Pattern.compile("^[0-9]{10}$");

Matcher matcher = pattern.matcher(phno);

boolean matchfound = matcher.find();

if(matchfound)

{

System.out.println("Validated! The phone number is a 10 digit number");

}

else

{

System.out.println("Invalid! The number is not a 10 digit number");

}

}

}
```

Output:



```
D:\LTI Work Related\Scenario\JavaScenarios>java RegexPrograms
Enter the phone number :
9819743560
Validated! The phone number is a 10 digit number
```

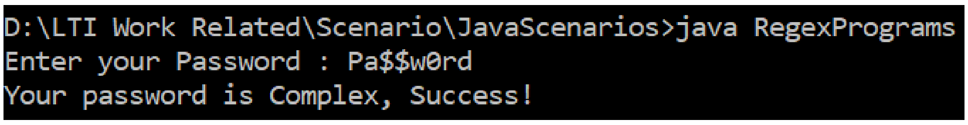

```
D:\LTI Work Related\Scenario\JavaScenarios>java RegexPrograms
Enter the phone number :
8885456
Invalid! The number is not a 10 digit number
```

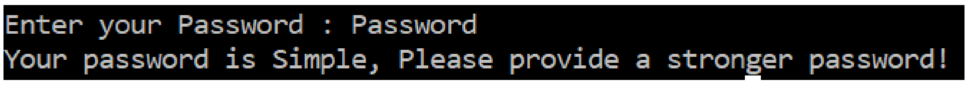
b. Password regex:

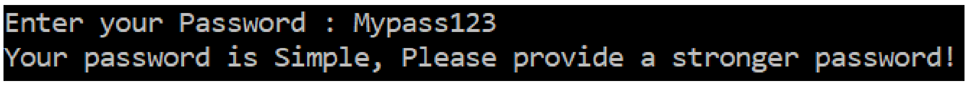
Code:

```
import java.util.*;

import java.util.regex.*;

public class RegexPrograms{


public static void main(String[] args)

{

Scanner sc= new Scanner(System.in);

System.out.print("Enter your Password : " );

String pass= sc.nextLine();

sc.close();

String reg = "^(?=.*[0-9])"+ "(?=.*[a-z])(?=.*[A-Z])"+"(?=.*[@#$%!])"+"(?=\\S+$).{7,15}$";

Pattern pattern = Pattern.compile(reg);

Matcher matcher = pattern.matcher(pass);

boolean matchfound = matcher.find();

if(matchfound)

{

System.out.println("Your password is Complex, Success!");

}

else

{

System.out.println("Your password is Simple, Please provide a stronger password!");

}

}

}
```

Output:


```
D:\LTI Work Related\Scenario\JavaScenarios>java RegexPrograms
Enter your Password : Pa$$w0rd
Your password is Complex, Success!
```


```
Enter your Password : Password
Your password is Simple, Please provide a stronger password!
```


```
Enter your Password : Mypass123
Your password is Simple, Please provide a stronger password!
```

Java Exception Handling

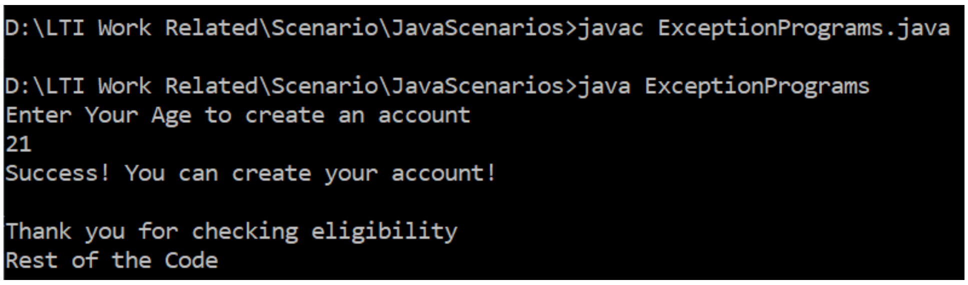
2.

Code:

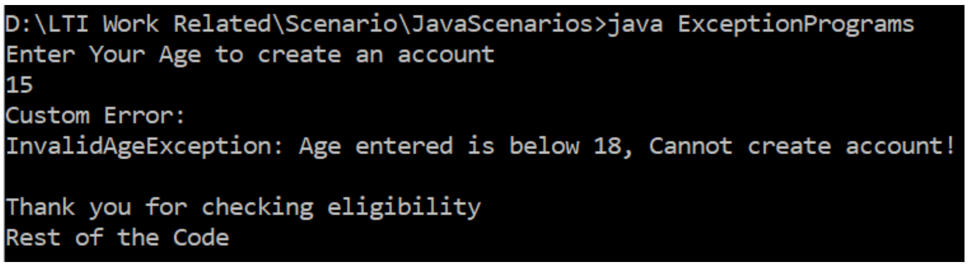
```
import java.util.*;


class InvalidAgeException extends Exception
{
   public InvalidAgeException (String str)
   {
      super(str);
   }
}


public class ExceptionPrograms{
   public static void checkUserAge(int age) throws InvalidAgeException{
      if(age < 18)
{
         throw new InvalidAgeException("Age entered is below 18, Cannot create account!");
      }
      else
{
         System.out.println("Success! You can create your account!");
      }
   }


   public static void main(String[] args)
{
      System.out.println("Enter Your Age to create an account");
      try
{
         Scanner sc = new Scanner(System.in);
```

```
            int age = Integer.parseInt(sc.nextLine());

            sc.close();

            checkUserAge(age);

        }

        catch(NumberFormatException e)

{

            System.out.println("Format Error: "+e);

        }


        catch(InvalidAgeException e)

{

            System.out.println("Custom Error: \n"+e);

        }

catch(Exception e)

{

System.out.println("Other Errors :"+e);

}

        finally{

            System.out.println("\nThank you for checking eligibility");

        }

        System.out.println("Rest of the Code");

    }

}
```

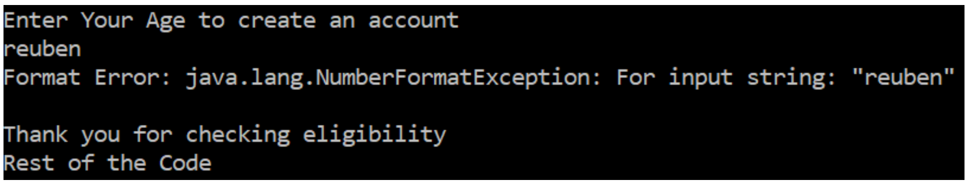
Output:



```
D:\LTI Work Related\Scenario\JavaScenarios>javac ExceptionPrograms.java

D:\LTI Work Related\Scenario\JavaScenarios>java ExceptionPrograms
Enter Your Age to create an account
21
Success! You can create your account!

Thank you for checking eligibility
Rest of the Code
```



```
D:\LTI Work Related\Scenario\JavaScenarios>java ExceptionPrograms
Enter Your Age to create an account
15
Custom Error:
InvalidAgeException: Age entered is below 18, Cannot create account!

Thank you for checking eligibility
Rest of the Code
```



```
Enter Your Age to create an account
reuben
Format Error: java.lang.NumberFormatException: For input string: "reuben"

Thank you for checking eligibility
Rest of the Code
```

Java OOPS

3

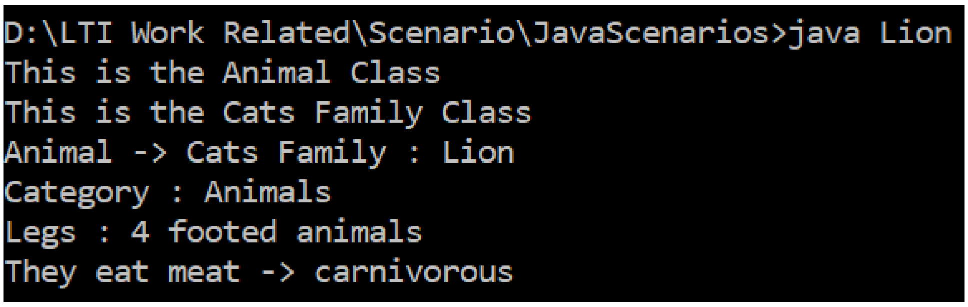
a. Multilevel Inheritance

Code:

```
class Animals{

   public Animals()

   {

    System.out.println("This is the Animal Class");

   }

   public void animalCategory()

   {

    System.out.println("Category : Animals");

   }

}


class CatsFamily extends Animals{

   public CatsFamily()

   {

    System.out.println("This is the Cats Family Class");

   }

   public void feetCount()
```

```
    {
     System.out.println("Legs : 4 footed animals");
    }
    public void eats()
    {
     System.out.println("They eat any food");
    }
}

public class Lion extends CatsFamily {
   public Lion()
   {
System.out.println("Animal -> Cats Family : Lion");
   }
   public void eats()
   {
System.out.println("They eat meat -> carnivorous");
   }
   public static void main(String args[])
   {
 Lion obj = new Lion();
 obj.animalCategory();
 obj.feetCount();
 obj.eats();
   }
}
```

Output:

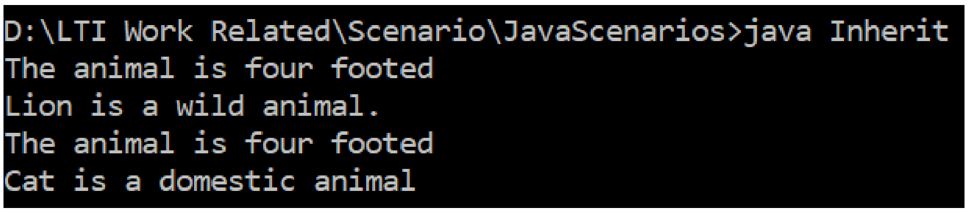

```
D:\LTI Work Related\Scenario\JavaScenarios>java Lion
This is the Animal Class
This is the Cats Family Class
Animal -> Cats Family : Lion
Category : Animals
Legs : 4 footed animals
They eat meat -> carnivorous
```

b. Hierarchical Inheritance

Code:

```
class Animal{

   public void feetCount(){

      System.out.println("The animal is four footed");

   }

}


class Lion extends Animal{

   public void lionMethod(){

      System.out.println("Lion is a wild animal.");

   }

}


class Cat extends Animal{

   public void catMethod(){

      System.out.println("Cat is a domestic animal");

   }

}


public class Inherit

{
   public static void main(String[] args)
{
       Lion l=new Lion();
       Cat c=new Cat();
       l.feetCount();
       l.lionMethod();
c.feetCount();
       c.catMethod();
   }
}
```

Output:



```
D:\LTI Work Related\Scenario\JavaScenarios>java Inherit
The animal is four footed
Lion is a wild animal.
The animal is four footed
Cat is a domestic animal
```

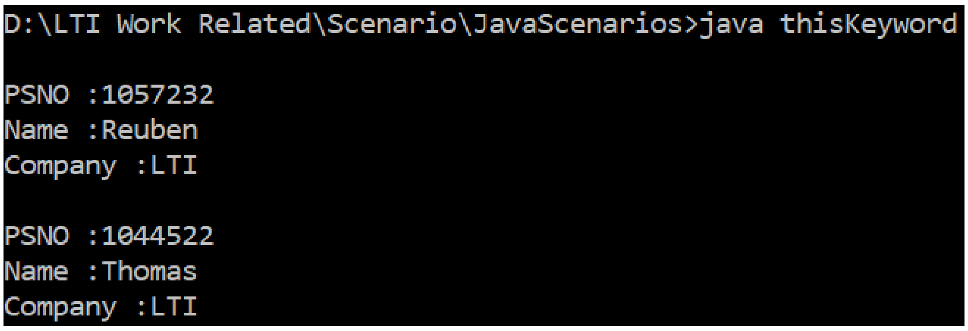
c. This keyword

Code:

```
class Trainee
{
int psno;
String name;
String company;
   Trainee(int psno,String name,String company)
{
this.psno = psno;
this.name = name;
this.company = company;
   }
void displayTrainee()
{
```

```
System.out.println("\nPSNO :" +psno+"\nName :"+name+"\nCompany :"+company);
}
}

public class thisKeyword
{
   public static void main(String[] args)
{
       Trainee t1 = new Trainee(1057232,"Reuben","LTI");
Trainee t2 = new Trainee(1044522,"Thomas","LTI");
t1.displayTrainee();
t2.displayTrainee();
   }
}
```

Output:



```
D:\LTI Work Related\Scenario\JavaScenarios>java thisKeyword

PSNO :1057232
Name :Reuben
Company :LTI

PSNO :1044522
Name :Thomas
Company :LTI
```