Lab-5

June 2, 2021

2b. Define one class A in package apack. In class A, four variables are defined of access
modifiers default, protected, private and public. Define class B in package bpack which
extends A and write display() method which access variables of class A. Define class C in package cpack which has one method display() in that create one object of class A and display its variables. Define class ProtectedDemo in package dpack which contains the main () method. Create objects of class B and C and call display method for
both these objects. Analyze the program by interpreting the access modifiers and provide
valid conclusion.

Code:

```
package mypack.apack;

public class a {
    int x;
    protected int y;
    private int z;
    public int w;
```

```
    public a(){
        x=1;
        y=2;
        z=3;
        w=4;
    }
```

```
}
```

```
package mypack.bpack;
```

```java
import mypack.apack.*;
public class b extends a{
    public void disp(){
        System.out.println("Class B");
        // System.out.println(" x :"+x);
        System.out.println("protected y :"+y);
        // System.out.println("z :"+z);
        System.out.println("public w :"+w);
    }
}
```

```java
package mypack.cpack;
import mypack.apack.*;
import mypack.bpack.*;
public class c {
    public void disp(){
        a obj = new a();
        System.out.println("Class C");
        // System.out.println("x : "+obj.x);
        // System.out.println("y : "+obj.y);
        // System.out.println("z : "+obj.z);
        System.out.println("public w : "+obj.w);
    }
}
```

```java
package mypack.dpack;
import mypack.apack.*;
import mypack.bpack.*;
import mypack.cpack.*;
public class protectedDemo {

    public static void main(String[] args) {
        b ob = new b();
        c oc = new c();

        ob.disp();
```

```java
        oc.disp();
    }
}
```

Output:

```
Class B
protected y :2
public w :4
Class C
public w : 4
PS C:\Users\Asus\Desktop\G
```

3 a. Consider a student examination database system that prints the mark sheet of the
students. Input the following from the command line student name and marks in 6 subjects. These marks should be in between 0 and 50 if the marks are not in the specified
range raise a Range Exception else find the total marks and print the percentage of the
student.

Code:

```
import org.w3c.dom.ranges.RangeException;

// import RangeException;
```

```
public class cmdExcep {

        public static void main(String args[]){
         String name = args[0];
         int m[] = new int[6];
         int k = 0;
         int sum=0;
        for(int i=1;i<7;i++) {

        //    if(int(args[i])>50 || args[i]<0)
         m[k]=Integer.parseInt(args[i]);
         k++;
        }
        for(int i=0;i<6;i++) {

            if(m[i]>50 || m[i]<0){
                  throw new ArithmeticException("invalid marks
 "+m[i]);
            }
            else{

                // for(int j=0;j<6;j++) {
```

```
                    //    if(int(args[i])>50 || args[i]<0)
                    //  m[i]=Integer.parseInt(args[i]);
                    sum+=m[i];
                    // }
                // System.out.println("Name :"+name);

                // System.out.println("percentage :"+(sum*100)
/300);

            }


            }
            System.out.println("Name :"+name);

            System.out.println("percentage :"+(sum*100)/300);

        // System.out.println(args[i]);

        }
        }
```

```
PS C:\Users\Asus\Desktop\GIT> cd codes/javacodes
PS C:\Users\Asus\Desktop\GIT\codes\javacodes> javac cmdExcep.java
PS C:\Users\Asus\Desktop\GIT\codes\javacodes> java cmdExcep.java Sun 45 50 50 48 49 50
Name :Sun
percentage :97
PS C:\Users\Asus\Desktop\GIT\codes\javacodes> java cmdExcep.java Sun 45 50 50 48 49 60
Exception in thread "main" java.lang.ArithmeticException: invalid marks 60
        at cmdExcep.main(cmdExcep.java:21)
PS C:\Users\Asus\Desktop\GIT\codes\javacodes>
```

Create a class temperature with member variable temp. Implement exception handling to test if temperature is equal to zero.

Code:

```
import java.util.Scanner;

public class tempExcep {
```

```
        static void checkTemp(int temp) {
          if (temp == 0) {
            throw new ArithmeticException("temperature is 0");
          }
          else {
            System.out.println("temperature is "+temp);
          }
        }

        public static void main(String[] args) {
          Scanner sc = new Scanner(System.in);
          int temp = sc.nextInt();
          checkTemp(temp);
          sc.close();
        }

}
```

Output:

```
25
temperature is 25
PS C:\Users\Asus\Desktop\GIT>  & 'c:\Users\Asus\.vscode\extensions\vscjava.vscode-jav
ncher.bat' 'C:\Program Files\Java\jdk-16\bin\java.exe' '--enable-preview' '-XX:+ShowC
ges' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\Asus\AppData\Roaming\Code\User\workspace
c9cbba4cea43d2\redhat.java\jdt_ws\GIT_d4ab13ff\bin' 'tempExcep'
0
Exception in thread "main" java.lang.ArithmeticException: temperature is 0
        at tempExcep.checkTemp(tempExcep.java:7)
        at tempExcep.main(tempExcep.java:18)
PS C:\Users\Asus\Desktop\GIT>
```

Implement a growable stack using interface and class. Interface will contain abstract methods push() and pop().

Code:

```
import java.util.Scanner;
import java.util.ArrayList;
interface operations
{
    void pop();
    void push(int a);
```

```
    void displayTop();
}
    class stack2 {
    ArrayList<Integer> stack=new ArrayList<Integer>();
    int currentSize=0;
    void push(int a)
    {
        stack.add(a);
        currentSize++;
    }
    void pop()
    {
        if(stack.isEmpty()==true)
        {
            System.out.println("stack is empty cant pop");
        }
        else
        {
            System.out.println(stack.get(currentSize-1)+" is p
oped");
            stack.remove(currentSize-1);
            currentSize--;
        }
    }
    void displayTop()
    {

        if(stack.isEmpty()==true)
        {
            System.out.println("stack is empty cant pop");
        }
        else
        {
            System.out.println(stack.get(0));
        }
    }
}
public class stack1
{
    public static void main(String[] args)
    {
        stack2 S=new stack2();
        Scanner sc=new Scanner(System.in);
        int x;
        do
        {
            System.out.println("enter 0 to exit");
            System.out.println("enter 1 to push");
            System.out.println("enter 2 to pop");
```

```
            System.out.println("enter 3 to display top element
");
            x=sc.nextInt();

            switch(x)
            {
                case 1 : int n;
                         System.out.println("enter element to
be pushed:");
                         n=sc.nextInt();
                         S.push(n);
                break;
                case 2 : S.pop();
                break;
                case 3 : S.displayTop();
                break;
                case 0 : break;
            }
        } while(x>0);
        sc.close();
    }
}
```

Output:

```
enter 0 to exit
enter 1 to push
enter 2 to pop
enter 3 to display top element
1
enter element to be pushed:
25
enter 0 to exit
enter 1 to push
enter 2 to pop
enter 3 to display top element
3
25
enter 0 to exit
enter 1 to push
enter 2 to pop
enter 3 to display top element
2
25 is poped
enter 0 to exit
enter 1 to push
enter 2 to pop
enter 3 to display top element
0
PS C:\Users\Asus\Desktop\GIT>
```