Name – Priyanshu Jhansla

Roll.no. - 88008

Ques.1. Create an Exception subclass UnderAge, which prints “Under Age” along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18. Write main() method also to show working of the program.

Ans.

```
public class UnderAge extends Exception {
 final private int age;
 public UnderAge(int age) {
 this.age = age;
 }
 @Override
 public String getMessage() {
 return "UnderAge: " + age + " is less than 18";
 }
}
/**** exceptionDemo.java ****/
import java.util.Scanner;
class exceptionDemo {
 static void test(int age) throws UnderAge {
 if (age < 18)
 throw new UnderAge(age);
 }
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter Age: ");
 int age = sc.nextInt();
 try {
 test(age);
 System.out.println("Test Successful");
 } catch (UnderAge e) {
 System.err.println(e.getMessage());
 System.out.println("Test Unsuccessful");
 } finally { sc.close();
 }
 }
}
```

Ques.5. Write a program to implement stack. Use exception handling to manage underflow and overflow conditions.

Ans.

```
/**** Stack.java ****/
```

```java
public class Stack {
 // Top of the Stack
 private int tos;
 // Array of Elements
 private int[] array;
 // Size of the Stack
 final private int size;
 /**
 * Public Constructor for Stack Objects
 *
 * @param size Size of the Stack
 */
 public Stack(int size) {
 this.tos = -1;
 this.size = size;
 this.array = new int[this.size];
 }
 /**
 * Push an element to the top of the stack
 *
 * @param e Element to be pushed
 * @throws StackException Stack Overflow
 */
 public void push(int e) throws StackException {
 if (tos == size - 1)
 throw new StackException("Stack Overflow: could not push " + e);
 else
 this.array[++this.tos] = e;
 } /**
 * Pop an element from the stack
 *
 * @return Object on top of the stack
 * @throws StackException Stack Underflow
 */
 public int pop() throws StackException {
 if (this.tos < 0) {
 throw new StackException("Stack Underflow: could not pop");
 } else
 return this.array[this.tos--];
 }
 /**
 * Index of the element at the top of the stack
 *
 * @return Index of Generic Element
 */
 public int getTOS() {
 return this.tos;
 }
```

```
 /**
 * Representation of Stack Object
 *
 * @return String Representation
 */
 @Override
 public String toString() {
 return "Stack<size=" + this.size + ">";
 }
}
/**** StackException.java ****/
public class StackException extends Exception {
 final private String message;
 public StackException(String message) {
 this.message = message;
 }
 @Override
 public String getMessage() {
 return this.message;
 }
}
/**** Main.java ****/
import java.util.Random;public class Main {
 public static void main(String[] args) {
 int r;
 Stack stack = new Stack(10);
 Random random = new Random(1337);
 System.out.println("Created stack of size 10...");
 System.out.println("Pushing integers onto stack...");
 while (true) {
 r = random.nextInt(100);
 System.out.println("Pushing " + r + "...");
 try {
 stack.push(r);
 System.out.println(
 "Elements in Stack = " + (stack.getTOS() + 1)
 );
 } catch (StackException e) {
 System.err.println(e.getMessage());
 break;
 }
 }
 System.out.println("Popping integers from stack...");
 while (true) {
 System.out.println(
 "Elements in Stack = " + (stack.getTOS() + 1)
 );
 try {
```

```
 System.out.println("Popped " + stack.pop() + "...");
 } catch (StackException e) {
 System.err.println(e.getMessage());
 break;
 }
 }
 }
}
```

Ques..6 Write a program that copies content of one file to another. Pass the names of the files through command-line arguments.

Ans.

```
/**** Copy.java ****/
import java.io.*;
public class Copy {
 public static void main(String[] args) throws Exception {
 if (args.length != 2) {
 System.err.println("Usage: java Copy <src> <dest>");
 } else {
 int i;
 FileInputStream fin = new FileInputStream(args[0]);
 FileOutputStream fout = new FileOutputStream(args[1]);
 while ((i = fin.read()) != -1) {
 fout.write(i);
 }
 fin.close();
 fout.close();
 System.out.println(
 "Copied contents of" + args[0] + " to " + args[1]
 );
 }
 }
}
```

Ques.7. Write a program to read a file and display only those lines that have the first two characters as ‘//’ (use try with resources)

Ans.

```
import java.io.*;
import java.util.*;
public class Read {
public static void main(String[] args) {
```

```
if (args.length != 1) {
System.err.println("Usage: java Read <src>");
}

else {
try (Scanner in = new Scanner(new File(args[0]))) {
while ( in .hasNext()) {
String str = in .nextLine();
if (str.substring(0, 2).equals("//"))
System.out.println(str);
}
} catch (IOException e) {
e.printStackTrace();
}
}
}
}
```