**Institute of Technology Tralee**

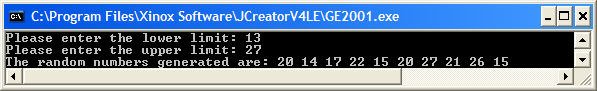
**Computing Department**

**Object Oriented Programming 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Session 3 – Methods in Java**

**(a)** Write a Java program that makes use of a user-defined method called **randomNumber**() that takes 2 integer arguments. The first argument is the lower limit and the second argument is the upper limit for the random number to be generated. The randomly generated number should be a whole number inclusive of these limits. The main() method should ask the user to supply the lower and upper limits and then call randomNumber() exactly 10 times (use a loop here!) to generate random numbers within these limits. The generated numbers should then be displayed to the console, separated by a blank as indicated in the following screenshot:

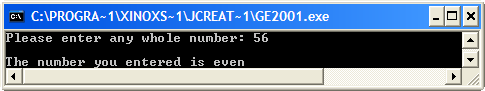


**(b)** Write a Java program that utilises a user-defined Java method called **isOdd**() that takes a single argument representing a **whole number**. The method should return the boolean value true or false depending on whether the number is odd or not. The main() method is to be written so that it asks the user to supply a whole number (no validation necessary here) and then calls isOdd() to determine whether the number is odd. The program will then display the outcome as indicated in the following sample screenshots:

**Run 1 – an odd number is entered**



**Run 2 – an even number is entered**



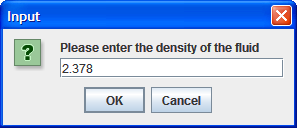
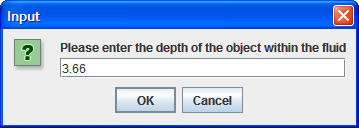
**(c)** The pressure on a body immersed in a fluid is given by the formula

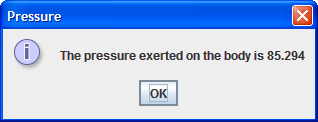
**Pressure = pgh**

where

* p is the density of the fluid,
* g is the acceleration due to earths gravity, which is 9.8 m/s/s and
* h is the depth of the body within the fluid

Write a Java program that utilises a user-defined method called **pressureCalculator**() that takes 2 floating-point arguments representing user-supplied values for the density of the fluid and the depth of the body. The method should return the pressure on the body based on the formula above. The main() method is to be written so that it asks the user to supply values for the density and depth and then calls pressureCalculator () to determine the pressure on the body. The pressure then gets displayed in main() to **3 decimal places**, as indicated in the sample screenshots below:

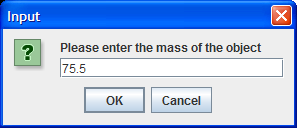
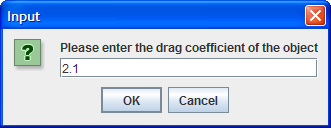


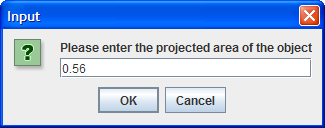
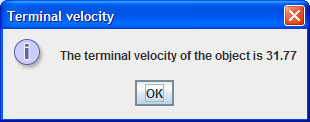
**(d)** A skydiver or parachutist falling from rest from a sufficient height above the earth will eventually reach a maximum speed called the terminal velocity, vt , which can be determined using the formula:

where

* m is the mass of the body
* g is the acceleration due to earth’s gravity, which is 9.8 m/s/s
* p is the air density, which you can take to be 1.247 kg/m3
* D is the drag coefficient
* A is the projected area of the object

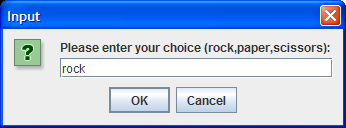
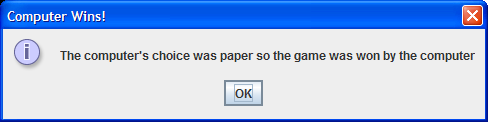
Write a Java program that utilises a user-defined method called **terminalVelocity**() that takes 3 floating-point arguments representing user-supplied values for the mass of the object, the drag coefficient and the projected area of the object. The method should return the terminal velocity of the object, based on the formula above. Your terminalVelocity() method should define p and g as **constants**. The main() method is to be written so that it asks the user to supply values for the mass, drag coefficient and projected area and then calls terminalVelocity () to determine the terminal velocity of the body. The terminal velocity then gets displayed in main() to **2 decimal places**, as indicated in the sample screenshots below:

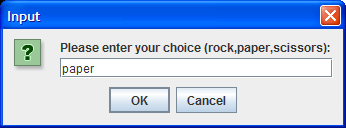
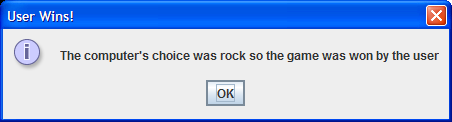
 

**(e)** Write a program that aims to simulate the game of rock, paper, scissors. The program will begin by asking the user to enter their choice of these 3 options in main(). Then a **random whole number** between 1 and 3 will be generated by a user-defined method **computerChoice**(). We will associate the number 1 with “rock”, 2 with “paper” and 3 with “scissors” in this case and the method will return one of these 3 words when it is called. Then, back in main(), the users choice will be compared with the computer’s choice, and the computer choice, along with the outcome of the game, will be displayed. In this game, “rock” beats “scissors”, “scissors” beats “paper” and “paper” beats “rock”. Your program requires **no loop** here as we will presume there is only one game played. It will run as indicated in the following sample screenshots:

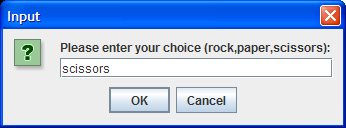
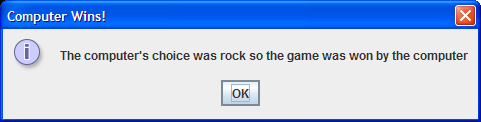
**Run 1:**

**Run 2:**

**Run 3:**

**Run 4:**

