INF 272

INFORMATIKA • DEPARTEMENT • INFORMATICS • DEPARTMENT
Universiteit van Pretoria
University of Pretoria

Practical 11 – Object oriented programming II

Out of 15 Marks

DUE: 2022-06-03 before 23:59

IMPORTANT NOTES:

- This is an individual micro practical.
- Practical work is based on assessment objectives.
- If an objective has been achieved a mark will be allocated.
- All micro practical activities conceptually (based on skills) feed into one or more homework assignments.
- By completing the practical work, you will learn the necessary skills to complete homework assignments.

INSTRUCTIONS:

- For this practical you will receive specific instructions that you need to follow.
- When uploading the practical homework, zip the entire project and then please upload it using the following naming convention. **Source Code:** Zip your source code files together and name it **uXXXXXXXX_PR11.zip**, where XXXXXXXX is your student number, e.g., u12345678_PR11.zip.

SUBMISSION DEADLINE: 2022-06-03 before 23:59

- There shall be no extensions to the deadline.
- If practical work submissions are uploaded too late, then upload errors WILL happen.
- Do not wait until the last minute to complete the practical assignments.
- Start working on the practical work as soon as it is published.
- Verify the completeness of your upload.
- There are multiple upload opportunities enabled if your upload is incomplete.
- Incomplete uploads will be considered unsubmitted work.
- E-mail submissions WILL NOT be accepted.
- Late submissions WILL NOT be accepted.
- **NO EXCEPTIONS WILL BE MADE FOR ANYONE.**

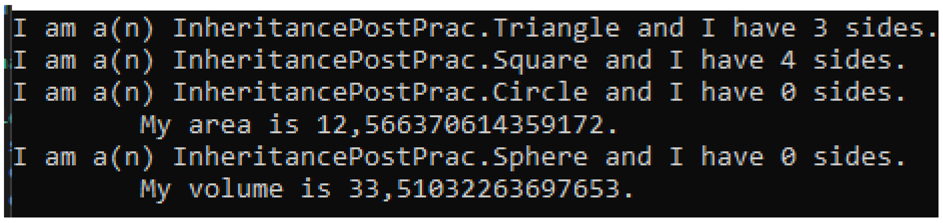
1. TOOLS, SKILLS, KNOWLEDGE REQUIRED:

- Object-oriented programming (OOP): Inheritance.

2. STANDARD REQUIREMENTS:

A game development studio is interested in building a console application that will be used by students to learn about shapes. The originator of the project sees it as something that will eventually be used to teach students about all kinds of shapes.

At the moment, you have been tasked with creating the first version of the application. The current version will only teach students about the triangle, square, circle, and sphere. When the application is run, it should show the output demonstrated in Figure 1 --- assuming we have a list that has a triangle, square, circle, and sphere (in that specific order). Since the project will eventually grow to include other things, you have been asked to use your knowledge of OOP, specially inheritance, to organise your code.



```
I am a(n) InheritancePostPrac.Triangle and I have 3 sides.
I am a(n) InheritancePostPrac.Square and I have 4 sides.
I am a(n) InheritancePostPrac.Circle and I have 0 sides.
        My area is 12,566370614359172.
I am a(n) InheritancePostPrac.Sphere and I have 0 sides.
        My volume is 33,51032263697653.
```

Figure 1: Example of the output to be generated using the getSides(), getArea(), and getVolume() methods

All shapes (i..e., triangle, square, circle, and cube) must have a **getSides()** method that returns the number of sides per shape. It must be possible to call such a method without knowing the exact type of shape that we are dealing with.

If we are dealing with a(n) instance of **Circle** that has a specific **radius**, it must be possible to call its associated **getArea()** method to retrieve its area. This method must only be available for circles and no other shape.

If we are dealing with a(n) instance of **Sphere** that has a specific **radius**, it must be possible to call its associated **getVolume()** method to retrieve its volume. This method must only be available for circles and no other shape.

Create all the necessary objects and methods, iterate over the list of objects and use the newly created methods to print out the text shown in Figure 1.

3. RESEARCH / READING REQUIREMENTS

- No additional research is required.
- All the content presented in, and / or may be found in the class example provides the knowledge and skills to complete this activity.

INF 272

Practical 11 – Object oriented programming II

Out of 15 Marks

DUE: 2022-06-03 before 23:59

Checklist	MAX
1. **Requirement 1 – Create a** getSides() **method for all shapes**	5
2. **Requirement 2 – Create a** getArea() **method in Circle only**	5
3. **Requirement 3 – Create a** getVolume() **method in Sphere only**	5
TOTAL MARK ALLOCATION	**15**