

## Assignment 6

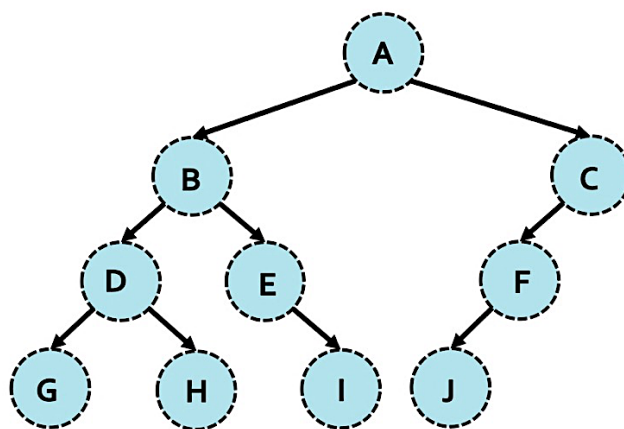
*Advanced Programming (INFO135)*

**Published at:** 13:00, Friday, 15.04.2022

**Deadline:** 13:00, Friday, 22.04.2022

---

1. Which of the following sequences determines the order in which the elements would be accessed during a **Pre-order** traversal:



- a) A B D G H E I C F J
  - b) A B C D E F G H I J
  - c) G H D I E B J F C A
  - d) G D H B E I A J F C
  - e) None of them
- 

2. Write a class called **QuizGift** that has a method `compute_result()` to solve the following problem and to compute the result. Then, write another method called `print_result()` that prints out the result.

*Sara is going to attend a written quiz where she can receive a prize based on the number of points she obtains. The written quiz has 5 questions each of them is worth different points and each takes different amount of time to answer. Sara will have 100 minutes and can choose which subset of questions to answer from the following question set:*

- Question 1 has 120 points and it takes 15 minutes to answer

- Question 2 has 200 points and it takes 20 minutes to answer
- Question 3 has 150 points and it takes 40 minutes to answer
- Question 4 has 350 points and it takes 50 minutes to answer
- Question 5 has 100 points and it takes 20 minutes to answer
- Question 6 has 90 points and it takes 10 minutes to answer

Sara will receive a watch if she obtains up to 250 points, a smartphone if she obtains 250 - 750 points, and, a laptop if she obtains more than 750 points. Sara would like to have a Python program, based on Dynamic Programming, to compute the maximum number of points she can obtain (in the given time) and to print it out. The program should also print the gift that she will receive as the result of answering the quiz. Please help her!

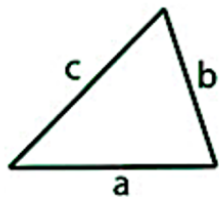
**Note:** You can write more methods in the **QuizGift** class if needed.

---

3. Write an **interface** called **Shape** that has an abstract method called `compute_area(self)`. Then, write the following classes, all of them *implementing* the Shape interface:

- a) **Square** class that has a constructor which receives as parameter the side of the square and sets it as an instance variable. This class should implement `compute_area()` method to compute and print the area of the square (area of square = side \* side)
- b) **Circle** class that has a constructor which receives as parameter radius of the circle and sets it as an instance variable. This class should implement `compute_area()` method to compute and print the area of the circle (area of circle = 3.14 \* radius \* radius)
- c) **Triangle** class that has a constructor which receives as parameters the values of the 3 sides of the triangle and sets them as instance variables. This class should implement `compute_area()` method to compute and print the area of triangle, based on the following formula:

$$s = \frac{a + b + c}{2}$$

$$\text{Area} = \sqrt{s(s - a)(s - b)(s - c)}$$


**Note:** check the examples of interface provided in the Lectures.

```
my_square = Square(2)
my_circle = Circle(2)
my_triangle = Triangle(5, 4, 3)

print('Area of square:', end=' ')
my_square.compute_area()
print('Area of circle:', end=' ')
my_circle.compute_area()
print('Area of triangle:', end=' ')
my_triangle.compute_area()
```

### Output:

```
Area of square: 4
Area of circle: 12.56
Area of triangle: 6.0
```

- 
4. Write a class called **House**. Each object of the **House** has an owner and a price value, and is in a certain condition (such as “Great”, “Good”, or “Bad”). House class has a **class variable** **count** that stores the total number of houses. The **House** class should also have the following methods:
- a constructor method that receives the parameter **owner**, **condition**, **price** and sets them up as instance variables. The constructor also initially sets two more variables: **cost** = 0, **sold** = False. When a new house is created, variable **count** is incremented by 1.
  - a method called **sell()** that receives the parameter **new\_owner** and sets it as the owner of house. It also sets the variable **sold** to True. Then, the method computes the profit which is equal to **price** - **cost** and prints it out.
  - a method called **change\_price()** that receives the parameter **new\_price**. If the house is already sold, it prints “House has been sold!”. Otherwise, it changes the value of the **price** to **new\_price**.

- a method called **renovate()** that receives the parameter `expense`, and `new_condition`. The method updates the `cost` value to `cost = cost + expense` and updates the value of `condition` to `new_condition`. Then, it prints out "House renovated!".
- a method **print\_info()** that prints the information related to a House, such as `owner`, `condition`, and `price` in dollar.

Create two objects of the House and call them `house1` and `house2`. Choose values of your choice for the parameters `owner` and a `price`, and `condition`.

Then, renovate `house1` and sell it by calling the appropriate methods. Then, print the information for both house objects, and finally, print the total number of houses. See the following output as an example:

**[Output]:**

```
Owner: John, Condition: Good, Price: 100000 [$]
Owner: Sara, Condition: Bad, Price: 250000 [$]
```

```
House renovated!
House sold! Profit: 50000
```

```
Owner: Leo, Condition: Great, Price: 100000 [$]
Total number of houses: 2
```