Syntax of creating a Function

A Function is a collection of statements that are grouped together to perform an operation.

Syntax:

```
1  public return_type function_name(parameter list)
2  {
3     //function body
4  }
```

- **Return type:** This defines which type of value function will return. If the return type is int then the function will return integer value. Your function can return String, double or any other valid data type. If you do not want to return any value then make it void.
- **Parameter list:** It is the type, order, and number of parameters (values) a function will take as input. Function may contain zero parameters (zero inputs).

Example 1

```
1  public void sayHello()                                  //no input
2  {
3     System.out.println("Hello");
4  }
5
6  public void sayHelloWithName(String name)               //String as input
7  {
8     System.out.println("Hello " + name);
9  }
```

Calling a function

Syntax:

```
function_name();
```

Example 2: Print Hello by using function [Using functions defined in example 1]

```
1  sayHello();
2  sayHelloWithName("Vikas");
```

Output:

```
Hello
Hello Vikas
```

Using function to create an object

```
1  class Item
2  {
3     String name;
4     int price;
5  }
```

Example 3: Use function to create objects

Functions:

```
1  public void createItem1()
2  {
3     Item item = new Item();
4     item.name = "Apple";
5     item.price = 5;
6     System.out.println("Item :: " + item.name + " : " + item.price);
7  }
8
9  public void createItem2(String name1, int price1)
10 {
11    Item item = new Item();
12    item.name = name1;
13    item.price = price1;
14    System.out.println("Item :: " + item.name + " : " + item.price);
15 }
```

Code:

```
1  createItem1();
2  createItem2("Pen", 2);
3  createItem2("Bag", 1000);
```

Output:

```
Item :: Apple : 5
Item :: Pen : 2
Item :: Bag : 1000
```

Problem 1	**Based on the class given below. Give answers for the code segments that follows.**
Vehicle	**Note: Explanation of each answer is mandatory.**
Class	

```
1  class Vehicle
2  {
3     int noOfTyres;
4     String type;
5  }
```

Problem 1.1	Refer the function given below and give output of code that follows.

Function

```
1  public void fn1()
2  {
3     Vehicle v = new Vehicle();
4     v.noOfTyres = 2;
5     v.type = "Bicycle";
6     System.out.println(v.noOfTyres);
7  }
```

Code

```
1  fn1();
```

Problem 1.2	Refer the function given below and give output of code that follows.

Function

```
1  public void fn2(String type1)
2  {
3     Vehicle v = new Vehicle();
4     v.noOfTyres = 4;
5     v.type = type1;
6     System.out.println("Vehicle :: " + v.type + " : " + v.noOfTyres);
7  }
```

Code

```
1  fn2("Car");
2  fn2("Truck");
```

Passing an object to functions

Example 4: Print an Item

Function:

```
1  public void print(Item item)
2  {
3     System.out.println("Item :: " + item.name + " : " + item.price);
4  }
```

Code:

```
1  Item it = new Item();
2  it.name = "Apple";
3  it.price = 20;
4  print(it);
```

Output:

```
Item :: Apple : 20
```

Passing multiple objects to functions

Example 5: Print the Item with less price

Function:

```
1  public void lowCost(Item itA, Item itB)
2  {
3     if(itA.price < itB.price)
4     {
5           System.out.println("Item1 :: " + itA.name + " : " + itA.price);
6     }
7     else
8     {
9           System.out.println("Item2 :: " + itB.name + " : " + itB.price);
10    }
11 }
```

Code:

```
1  Item it1 = new Item();
2  it1.name = "Apple";
3  it1.price = 20;
4  Item it2 = new Item();
5  it2.name = "Oranges";
6  it2.price = 15;
7  lowCost(it1, it2);
```

Output:

```
Item2 :: Oranges : 15
```

Functions that return object

Let's define a class named Vehicle with type (String) and brand (String) as its variables. We'll use a function to create an object of this class.

```
1  class Vehicle
2  {
3     String type;
4     String brand;
5  }
```

Example 6: Returning object from a function

Function:

```
1  public Vehicle makeVehicle(String type1, String brand1)
2  {
3     Vehicle v = new Vehicle();
4     v.type = type1;
```

```
5      v.brand = brand1;
6      return v;
7   }
```

Code:

```
1   Vehicle myCar;
2   myCar = makeVehicle("Car", "Hyundai");
3   Vehicle vehicle = makeVehicle("Truck", "Tata");
4   System.out.println("Vehicle :: " + myCar.type + " : " + myCar.brand);
5   System.out.println("Vehicle :: " + vehicle.type + " : " + vehicle.brand);
```

Output:

```
Vehicle :: Car : Hyundai
Vehicle :: Truck : Tata
```

Problem 2	**Based on the class given below. Give answers for the code segments that follows.**
Vehicle2	**Note: Explanation of each answer is mandatory.**

Class

```
1   class Vehicle2
2   {
3      int noOfTyres;
4      String type;
5   }
```

Function

```
1   public Vehicle2 fn3(String type1)
2   {
3      Vehicle2 v = new Vehicle2();
4      v.noOfTyres = 4;
5      v.type = type1;
6      return v;
7   }
```

Code

```
1   Vehicle2 v1 = fn3("Car");
2   Vehicle2 v2 = new Vehicle2();
3   v2.type = "E - Rickshaw";
4   v2.noOfTyres = 3;
5   Vehicle2 v3 = fn3("Auto Rickshaw");
6   v3.noOfTyres = 3;
7   System.out.println(v1.type + " : " v1.noOfTyres);
8   System.out.println(v2.type + " : " v2.noOfTyres);
9   System.out.println(v3.type + " : " v3.noOfTyres);
```

Problem 3	Based on the class given below. Give answers for the code segments that follows.
Item	Note: Explanation of each answer is mandatory.

Class

```
1  class Item
2  {
3      int price;
4      int quantity;
5      String name;
6  }
```

Problem 3.1	Refer the function given below and give output of code that follows.

Function

```
1  public Item makeItem(String name1, int price1, int quantity1)
2  {
3      Item item = new Item();
4      item.price = price1;
5      item.quantity = quantity1;
6      item.name = name1;
7      return item;
8  }
```

Code

```
1  Item item1 = makeItem("Apple", 20, 5);
2  System.out.println(item1.quantity + " : " + item1.name);
```

Problem 3.2	Refer the function given below and give output of code that follows.

Function

```
1  public void compareTotalCost(Item item1, Item item2)
2  {
3      if((item1.quantity * item1.price) > (item2.quantity * item2.price))
4      {
5             System.out.println(item1.name + "costs more");
6      }
7      else
8      {
9             System.out.println(item2.name + "costs more");
10     }
11
12 }
```

Code

```
1  Item it1 = makeItem("Apple", 20, 5);
2  Item it2 = makeItem("Orange", 26, 4);
3  compareTotalCost(it1, it2);
```

Problem 4.1	Create a class Point that takes two variables x and y of int type.
Point	
Problem 4.2	Complete the function makePoint() given below. makePoint() takes two ints and returns a Point object with values of x and y initialized with input values.

```
public Point makePoint(int x1, int y1)
{
   //complete this function
}
```

Problem 4.3	Complete the function printPoint() given below. printPoint() takes Point object as input and print the x and y value of this object on screen.

```
public void printPoint(Point p)
{
      //complete this function
}
```

Problem 4.4	Use makePoint() function, created in Problem 4.2, to create a new Point class object. Use random values to initialize variables x and y. Use printPoint() function, created in Problem 4.3, to print the object created in last step.

Problem 5.1	Create a class Student having variables name of type String and marks of type int.
Student	
Problem 5.2	Complete the function betterStudent() given below. betterStudent() that takes two Student objects as input and prints the student's name whose marks are better.

```
public void betterStudent(Student s1, Student s2)
{
      //complete this function
}
```

Problem 5.3	Create a Student with name Vikas and marks 90.
Problem 5.4	Create another student with name and marks of your choice.
Problem 5.5	Use betterStudent() function (with Student objects created in the previous steps) and give the output.