**Object Oriented Programming**

**Tutorial 01**

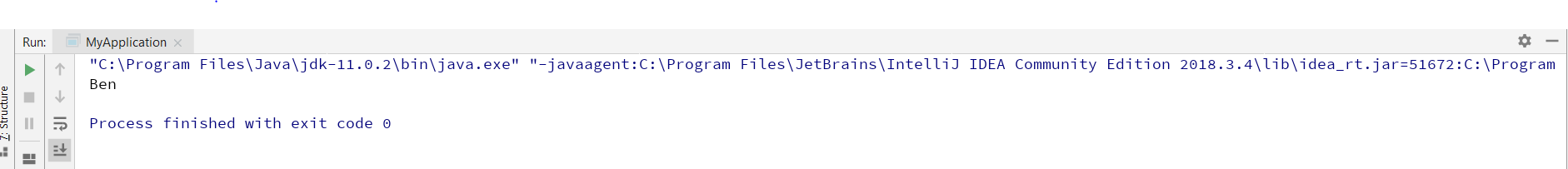
3) Let’s make the Person class do something: make constructor that takes a String and sets the name to be that String. Also, write a method called “displayName”, which will print the name out.

**package** iit.tutorial1.myapplication;  
  
**public class** Person {  
 **private** String **name**;  
 **private** String **sureName**;  
 **private int age**;  
  
 *//Constructor of the person class* **public** Person(String n){  
 **name** = n;  
 }  
  
 *//display name* **public void** displayName(){  
 System.***out***.println(**name**);  
 }

}

4) Let’s go back to our MyApplication class. Since we now have the Person class, let’s make some use out of it. In the "main" method of our MyApplication class, create a Person named "Ben".

**package** iit.tutorial1.myapplication;  
  
**public class** MyApplication {  
  
  
 **public static void** main(String[] args) {  
  
 Person obj = **new** Person(**"Ben"**);  
 obj.displayName();  
 }  
}



5) Write for the class Person

- void setSurname(Strin s)

- void setAge(int num);

**public void** setSureName(String s){  
 **sureName** = s;  
}  
  
**public void** setAge(**int** num){  
 **age** = num;  
  
}

6) Implement also accessory methods to return the surname and the age.

- String getSurname();

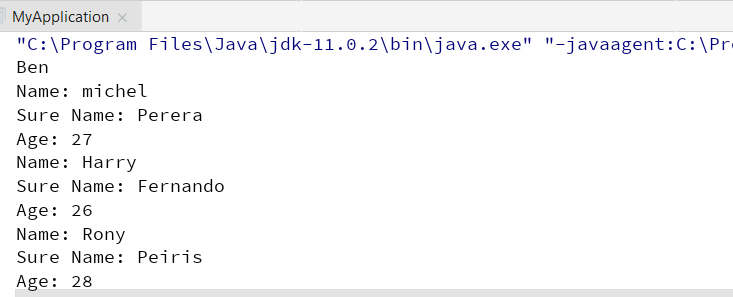
- int getAge();

**public** String getSureName(){  
 **return sureName**;  
}  
  
**public int** getAge(){  
 **return age**;  
}

7) In the main on MyApplication class create other three person and for all of them print on the screen their name , surname and age.

*// display details* **public void** printDetails(){  
 System.***out***.println(**"Name: "** + **name**);  
 System.***out***.println(**"Sure Name: "** + **sureName**);  
 System.***out***.println(**"Age: "** + **age**);

**package** iit.tutorial1.myapplication;  
  
**public class** MyApplication {  
  
  
 **public static void** main(String[] args) {  
  
 Person obj = **new** Person(**"Ben"**);  
 obj.displayName();  
  
 Person obj1 = **new** Person(**"michel"**,**"Perera"**,27);  
 Person obj2 = **new** Person(**"Harry"**,**"Fernando"**,26);  
 Person obj3 = **new** Person(**"Rony"**,**"Peiris"**,28);  
  
 obj1.printDetails();  
 obj2.printDetails();  
 obj3.printDetails();  
 }



9)

**package** iit.circle.tutorial1;  
  
**public class** Circle {  
 **private** String **color**;  
 **private double radius**;  
  
 *// default constructor* **public** Circle(){  
 **radius** = 1;  
 **color** = **"Blue"**;  
 }  
 *// Constructor of the circle class with one parameter* **public** Circle(**double** r){  
 **radius** = r;  
 **color** = **"Blue"**;  
 }  
  
  
 **public double** getRadius(){  
 **return radius**;  
 }  
  
 *// Calculate circle area* **public double** getArea(){  
 **return radius**\***radius**\*Math.***PI***;  
 }  
}

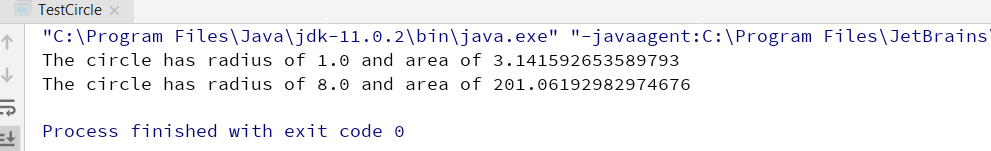
10) Can you run the Circle class?

I can’t rub a circle class.

Because, circle class doesn’t have main method.

11)

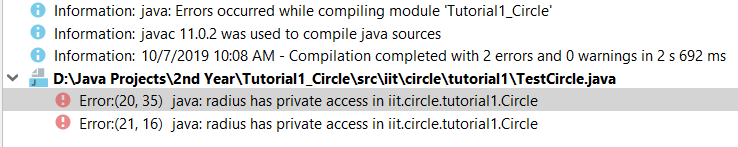
**package** iit.circle.tutorial1;  
  
**public class** TestCircle {  
   
 **public static void** main(String[] args) {  
 *// Declare an instance of Circle class called circle1  
 // Invoke the default constructor* Circle circle1 = **new** Circle();  
   
 *// invoke public methods* System.***out***.println(**"The circle has radius of "** + circle1.getRadius() + **" and area of "** + circle1.getArea());  
   
 *//declare an instance of Circle class called circle2  
 // invoke the second constructor* Circle circle2 = **new** Circle(8);  
   
 *// invoke public methods* System.***out***.println(**"The circle has radius of "** + circle2.getRadius() + **" and area of "** + circle2.getArea());  
 }  
}



13) In TestCircle, are you able to access the instance variable radius or to assigna new value to it directly?

Try the following line of code and explain why you get and error message: System.out.println(circle1.radius);

circle1.radius = 10;



Radius is private variable we cannot access directly from main

14)

*// Constructor of the circle class with two parameter***public** Circle(**double** r, String c){  
 **radius** = r;  
 **color** =c;  
}

15)

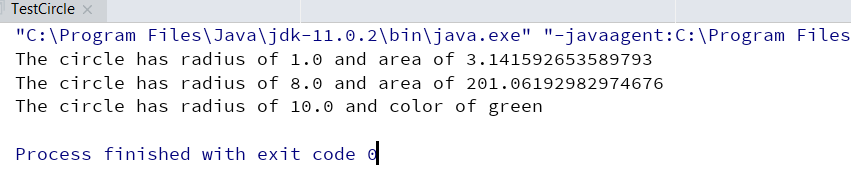
*//declare an instance of Circle class called circle3  
// invoke the third constructor*Circle circle3 = **new** Circle(10,**"green"**);

16)

**package** iit.circle.tutorial1;  
  
**public class** Circle {  
 **private** String **color**;  
 **private double radius**;  
  
 *// default constructor* **public** Circle(){  
 **radius** = 1;  
 **color** = **"Blue"**;  
 }  
 *// Constructor of the circle class with one parameter* **public** Circle(**double** r){  
 **radius** = r;  
 **color** = **"Blue"**;  
 }  
  
 *// Constructor of the circle class with two parameter* **public** Circle(**double** r, String c){  
 **radius** = r;  
 **color** =c;  
 }  
  
 **public double** getRadius(){  
 **return radius**;  
 }  
  
 *// Calculate circle area* **public double** getArea(){  
 **return radius**\***radius**\*Math.***PI***;  
 }  
  
 **public** String getColor(){  
 **return color**;  
 }

17)

**package** iit.circle.tutorial1;  
  
**public class** TestCircle {  
  
 **public static void** main(String[] args) {  
 *// Declare an instance of Circle class called circle1  
 // Invoke the default constructor* Circle circle1 = **new** Circle();  
  
 *// invoke public methods* System.***out***.println(**"The circle has radius of "** + circle1.getRadius() + **" and area of "** + circle1.getArea());  
  
 *//declare an instance of Circle class called circle2  
 // invoke the second constructor* Circle circle2 = **new** Circle(8);  
  
 *// invoke public methods* System.***out***.println(**"The circle has radius of "** + circle2.getRadius() + **" and area of "** + circle2.getArea());  
  
  
 *//declare an instance of Circle class called circle3  
 // invoke the third constructor* Circle circle3 = **new** Circle(10,**"green"**);  
  
 System.***out***.println(**"The circle has radius of "** + circle3.getRadius() + **" and color of "** + circle3.getColor());  
  
  
 }  
}



19)

**public void** setRadius(**double** newRadius){  
 **radius** = newRadius;  
}  
**public void** setColour(String newColour){  
 **color** = newColour;  
}

20) Can you run the following?

System.out.println(circle1.setRadius(20));

I can’t run this statement .

Because, setRadius method doesn’t have return type.

21) Modify all the constructors and the setter methods in the class Circle in order to use the keyword this.

**package** iit.circle.tutorial1;  
  
**public class** Circle {  
 **private** String **color**;  
 **private double radius**;  
  
 *// default constructor* **public** Circle(){  
 **radius** = 1;  
 **color** = **"Blue"**;  
 }  
 *// Constructor of the circle class with one parameter* **public** Circle(**double** radius){  
 **this**.**radius** = radius;  
 **color** = **"Blue"**;  
 }  
  
 *// Constructor of the circle class with two parameter* **public** Circle(**double** radius, String color){  
 **this**.**radius** = radius;  
 **this**.**color** =color;  
 }  
  
 **public double** getRadius(){  
 **return radius**;  
 }  
  
 *// Calculate circle area* **public double** getArea(){  
 **return radius**\***radius**\*Math.***PI***;  
 }  
  
 **public** String getColor(){  
 **return color**;  
 }  
  
 **public void** setRadius(**double** radius){  
 **this**.**radius** = radius;  
 }  
 **public void** setColour(String color){  
 **this**.**color** = color;  
 }  
}

23)