
# Gabriella Rakgotsoka

20232605

**JD521_Formative 1** **08/09/2023**

**No table of contents entries found.**

# Question 1

**Instructions: Code snippets below contain errors, you are required to find the error and re-write the code and fix the error. Compile it and provide the correct code snippet. For each correct code provided 5 Marks will be added**

**1. The following program has several errors. Modify it so that it will compile and run without**

**errors. [5 + 5 marks]**

```
// Filename: Temperature.java

PUBLIC CLASS temperature {

PUBLIC void main(string args) {

double fahrenheit = 62.5;

*/ Convert /*

double celsius = f2c(fahrenheit);

System.out.println(fahrenheit + 'F = ' + celsius + 'C');

}

double f2c(float fahr) {

RETURN (fahr - 32) * 5 / 9;

}

}
```

## Modification:

```
public class Temperature {
 // PUBLIC, CLASS are not keywords. The Filename should be same as the class name
        Run main | Debug main
        public static void main (String[] args) {
 // it is either static public void main or final public static void main and String[] args or String args[]
            double fahrenheit = 62.5;
           /* Convert */      // */  */ is not a comment
            double celsius = f2c( (float)fahrenheit);
            /* The method f2c(float) in the type Temperature is no apptlicable for the arguments (double)
           so it mudt be typecasted to float. The method f2c(float) in the type Temperature is not applicable for the
             arguments (double)*/
            System.out.println(fahrenheit + "F =" + celsius + "C");  // 'F=' isn't a character and cannot be
            // Concatenated
        }

        static double f2c(float fahr) {
            return (fahr - 32) * 5/9; // RETURN is not a keyword
        }
 }
```

**2. The following program has several errors. Modify it so that it will compile and run without errors. [3 + 5 Marks]**

```
public class MyClass extends Thread {
public MyClass(String s) { msg = s; }
String msg;
public void run() {
System.out.println(msg);
}
public static int class (String[] args) {
New MyClass("Hello");
new MyClass("World");
}
}
```

## Modifiaction:

```
public class MyClass extends Thread {

public MyClass(String s) { msg = s; }

String msg;

public void run() {

System.out.println(msg);

}

public class class{
    
public static int class (String[] args) {
//An identifier should be included

System.out.println("Hello");
//missing a method body
//illegal start of type

System.out.println("World");
//invalid method declaration type
//illegal start of type

}
}
 


}
```

**3. The following program has several errors. Modify it so that it will compile and run without**

errors. [5 + 5 Marks]

```
// filename Main.java
class Grandparent {
public void Print() {
System.out.println("Grandparent's Print()");
}
}
class Parent extends main {
public void Print(Grandparent) {
System.out.println("Parent's Print()");
}
}
class Child extends Parent {
public void Print() {
super.super.Print();
System.out.println("Child's Print()");
}
}
public class Main {
public static void main(String[] args) {
Parent c = new Child();
c.Print();
}
}
```

## Modification:

```java
// filename Main.java
class Grandparent {
    public void Print() {
        System.out.println("Grandparent's Print()");
    }
}

class Parent extends Grandparent {
    public void Print() {
        super.Print();
        System.out.println("Parent's Print()");
    }
}

class Child extends Parent {
    public void Print() {
        super.Print();
        //Compiler Error in super.super.Print()
        //We can only access Grandparent’s members using Parent.
        System.out.println("Child's Print()");
    }
}

class Main {
    Run main | Debug main
    public static void main(String[] args) {
        Child c = new Child();
        c.Print();
    }
}
```

**4. The following program has several errors. Modify it so that it will compile and run without errors. [4 +5 Marks]**

class Base {

public void show() {

System.out.println("Base::show() called");

}

}

class Derived extends main {

public void show() {

System.out.println("Derived::show() called");

}

}

public class Main {

public static void base(String[] args) {

Base b = new Derived();;

b.show();

}

}

## Modification:

```
class Base {

public void show() {

System.out.println("Base::show() called");

}

}

class Derived extends Base {
    //show() in Derived cannot override show() in Base
    //b is a referrence of Base type and refers to an object of derived
    public void show() {
System.out.println("Derived::show() called");

}

}

class Main {
    //file methods cannot be overridden

public static void base(String[] args) {

    Base b = new Derived();;

    b.show();

}

}
```

## Question 2

**1. What feature(s) of this code tells you that dynamic binding is taking place? [2 Marks]**

Polymorphism and Parent/super class reference that refer to subclass objects.

parkingLot.park(chevy);

parkingLot.park(honda); ... etc

**2. What is the output from main() ? [2 Marks]**

Vehicle with 4 wheels
Vehicle with 2 wheels
Vehicle with 3 wheels
Vehicle with 4 wheels

**3. True/False -- The Vehicle class is an abstract class. [2 Marks]**

False. The Vehicle class is a super class

**4. True/False -- Motorcycle is derived from Vehicle. [2 Marks]**

True

**5. True/False -- Car is derived from Vehicle. [2 Marks]**

True

**6. True/False -- setWheels() cannot be used polymorphically. [2 Marks]**

True

**7. When a Car object is created, which constructors are invoked, and in what order? [2 Marks]**

1st Car( ) and then Vehicle(int nrWheels)

**8. What is the purpose of the instance variable nrVehicles in the Lot class? [2 Marks]**

This is to keep track of the number of Vehicles parked

**9. Identify the common coding mistake in Lot's park method. How would you correct this coding error?[2 Marks]**

There is a lack of range checking when you park more than `nrVehicles` cars.

Modification:

```
public void park ( Vehicle v ) {
    if(nrVehicles < MAX_VEHICLES) {
      vehicles[nrVehicles++] = v;
    } else {
```

```
        throw new IllegalStateException("Lot full, go find another one!");
    }
}
```

**10. In the Vehicle constructor, what is the purpose of { this( 4 ); } ? [2 Marks]**

The purpose of this keyword is to eliminate the confusion between class attributes and parameters with the same name

**11. In the MotorCycle constructor, what is the purpose of { super( nrWheels); }? [2 Marks]**

Super(wheels) will set the number of wheels for the Motorcyle object, i.e. it will initialize the nrWheels variable in the Vehicle class to the specified value.

**12. Car's toString() method invokes getWheels(), even though getWheels( ) is not defined in the Car class. How is this possible? [2]**

This is due to inheritance, because Car is derived/extended from Vehicle class.

# Completed Declaration of Authenticity

I Gabriella Rakgotsoka _ hereby
(FULL NAME)

declare that the contents of this assignment JD521_FA2 is entirely my own work except for the following documents: (List the documents and page numbers of work in this portfolio
that were generated in a group)

| **Activity** | **Date** |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |

Signature: Date: 2022/09/08