

MANTHLY TEST JULY 2017

QUESTIONS FOR AVERAGE BLOOMERS

1. How many types of polymorphism?

Ans- 1.Static Polymorphism (compile time polymorphism/ Method overloading)
2.Dynamic Polymorphism (run time polymorphism/ Method Overriding)

2. What do you mean by Abstraction?

Ans - Abstraction in Object Oriented Programming refers to the ability to make a class abstract.

3. Which type of inheritance java support?

Ans – Multiple only

4. How one can make a text field un-editable on a frame ?

Ans- `(jTextField1.setEditable(false)`

5. How many types of containers can a java GUI application contain?

Ans- A java GUI application can contain as many containers as it needs but with a condition that it must have atleast one top level container.

6. What is the use of Final Keyword in Java?

Ans – The **final keyword** in java is used to restrict the user.

7. What do you understand by packages?

Ans – - Packages are used in Java in order to prevent naming conflicts, to control access, to make searching/locating and usage of classes, interfaces, enumerations and annotations easier, etc.

8. Name any two packages.

Ans- `java.lang` , `java.io`

9. What is method?

Ans – A **method** is a subprogram that acts on data and often returns a value.

10. What is the use of access specifier?

Ans – To set access levels for classes, variables, methods and constructors.

Questions for Brilliant achiever

1. What is Static Polymorphism ?

Ans - The ability to execute different method implementations by altering the argument used with the method name is known as method overloading.

2. What is Dynamic Polymorphism /run time polymorphism/ Method Overriding?

Ans - When you create a subclass by extending an existing class, the new subclass contains data and methods that were defined in the original superclass. In other words, any child class object has all the attributes of its parent.

3. Which keyword is used for inheritance in JAVA ?

Ans – In java *extends* keyword is used for inheritance.

4. What is the use of final keyword, explain?

Ans – The **final keyword** in java is used to restrict the user. The java final keyword can be used in many context. Final can be:

1. variable
2. method
3. class

The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only.

5. Give an example of packages?

Ans - /* File name : Animal.java */

```
package animals;

interface Animal {

    public void eat();

    public void travel(); }
```

6. What is the use of access specifier ? Explain.

Ans - Java provides a number of access modifiers to set access levels for classes, variables, methods and constructors. The four access levels are:

- Visible to the package. the default. No modifiers are needed.
- Visible to the class only (private).
- Visible to the world (public).
- Visible to the package and all subclasses (protected).

7. What is default access specifier?

Ans- Default access modifier means we do not explicitly declare an access modifier for a class, field, method, etc. A variable or method declared without any

access control modifier is available to any other class in the same package. The fields in an interface are implicitly public static final and the methods in an interface are by default public.

8. Can a variable which is declared private accessed outside ? Explain with reason.

Ans- Yes, Variables that are declared private can be accessed outside the class if public getter methods are present in the class.

9. Give an example of private access specifier?

```
Ans- public class Logger {  
  
    private String format;  
  
    public String getFormat() {  
  
        return this.format;  
  
    }  
  
    public void setFormat(String format) {  
  
        this.format = format;  
  
    }  
  
}
```

Here, the *format* variable of the Logger class is private, so there's no way for other classes to retrieve or set its value directly.

10. Explain protected access specifier?

Ans – The protected access modifier cannot be applied to class and interfaces. Methods, fields can be declared protected, however methods and fields in a interface cannot be declared protected. Protected access gives the subclass a chance to use the helper method or variable, while preventing a nonrelated class from trying to use it.

Example:

The following parent class uses protected access control, to allow its child class override *openSpeaker()* method:

```
class AudioPlayer {  
  
    protected boolean openSpeaker(Speaker sp) {  
  
        // implementation details  
  
    }  
  
}
```

```
class StreamingAudioPlayer {  
  
    boolean openSpeaker(Speaker sp) {  
  
        // implementation details  
  
    }  
  
}
```

Here, if we define *openSpeaker()* method as private, then it would not be accessible from any other class other than *AudioPlayer*. If we define it as public, then it would become accessible to all the outside world. But our intension is to expose this method to its subclass only, thats why we used *protected* modifier.

Question for late bloomers

1. What is encapsulation?

Ans - It refers to binding the data and its associated methods together in a single unit .

2. What is Polymorphism ?

Ans – polymorphism refers to do many task.

3. What is Inheritance?

Ans- The process of deriving multiple functions from one class to another.

4. What is class?

Ans – A class is a template for objects.

5. What is object?

Ans – An **object** is an instance of a class.

6. What is sub class ?

Ans- The derived class is a **subclass** or a child class.

7. What is message ?

Ans – A message is the information sent to the application.

8. What is super class ?

Ans – The original class that is used to derive a new class is called a **superclass**.

9. What command do you need to write in actionPerformed() event handler of a button, in order to make aexit button ?

Ans- System.exit(0);

10. How can you make a Text Field un-editable on a frame ?

Ans- jTextField1.setEditable(false);