1. **Which members of the Circle class are encapsulated?**

The radius and PI value are encapsulated because they are private and only the class can access it, allowing the program to keep it secure against hackers and unknown behaviours from the code.

2. **What name must the constructor of a class have**

A constructor should have the same name as the class so it can be accessed by the machine

3. **Explain the difference between private and public access modifiers**

The difference between private and public classes (or members) is that private ones can only be accessed within the same class, which keeps them secure and hidden from other parts of the program. This prevents other classes from changing or misusing the data. Public classes or members, on the other hand, can be accessed by any other class, meaning they are open for use throughout the program. Because of this, public members can be changed or called from outside the class. Using private helps protect data, while public allows controlled access when needed.

4. **Consider the following code. Is the last statement valid or invalid? Explain.**

```
Circle dot = new Circle(2);
dot.radius=5;
```

This code can be valid or invalid depending on whether the radius is public or private. If radius is public, then dot.radius can be changed, so the code is valid. If radius is private, then it cannot be accessed or modified from outside the class, which makes the code invalid. Based on the code shown, and since no access modifier is specified, it should be considered valid.

5. **Use the following class to answer the question**

```
Public class Roo {

Private int x;

Public Roo {
x=1;
}

Public void setX(int z) {
x=z;
}
```

```
Public int getX() {
return(x):
}

Publix int calculate() {
x= x*factor();
return(x):
{
Private int factor() {
return(0,12);
  }
}
```

a) **What is the name of the class?**

The name of the class is Roo

b) What is the name of the data member?

The name of the data member is called x

c) List the accessor method

The accessor method is getX

d) List the modifier method

The modifier method is setX

e) List the helper method

The helper method is factor

f) What is the name of the constructor?

The constructor name is Roo

g) How many methods members are there?

There are four method members

6. **What is the difference between a class and an object?**

A class is a blueprint that defines attributes and behaviors, while an object is an instance of that class that actually stores data and uses those behaviors.

9. Use the following class data member definitions to answer the questions below:

Public class Moo {

Private double y;
Private static int x;
Private static final z;

a) **Which data member is a constant**

Z is the constant data member because it is finalized and can not be changed

b) **Which data members are variables**

X and y can be changed because they are not finalized

c) **Which data members are instance members**

Y is the only instance member

d) **Which data members are class members**

X and Z are class members

11. Compare and contrast overriding methods to overloading methods.

Overloading and overriding are both ways to define methods, but they serve different purposes. Overloading happens within the same class when two or more methods share the same name but have different parameter lists, allowing multiple ways to perform a similar task. The return type can change, and the compiler decides which version to call based on the arguments. Overriding, however, happens between a superclass and a subclass, where the subclass provides a new version of a method that has the exact same name, parameters, and return type as the one in the parent class. This allows the subclass to change or specialize the behavior inherited from its parent. Overloading provides flexibility, while overriding provides customized behavior.