TK 1143 Program Design 2020/2021

Polymorphism

Section A:

1. Based on the following code, can you identify either it is constructor overloaded or method overloaded? Justify your answer.

```
public class Kira
{
  public Kira(){
    //-----> (1)
  }

  Public void Kira() {
     //-----> (2)
  }
}
```

Answer:

Constructor Overloading.
This is because there is constructors having the same name with different parameter.

2. a) Understand the following code and predict the output

Dynamic Polymorphism

```
//Class SeaCreature
public class SeaCreature {
    public void method1() {
          System.out.println("creature 1");
    }
    public void method2() {
          System.out.println("creature 2");
    }
    public String toString() {
          return "ocean dwelling";
    }
}
// Class Mammal
public class Mammal extends SeaCreature {
    public void method1() {
          System.out.println("warm blooded");
    }
}
// Class Whale
public class Whale extends Mammal {
    public void method1() {
            System.out.println("spout");
        }
        public String toString() {
            return "BIG!";
        }
    }
    // Class Squid
    public class Squid extends SeaCreature {
        public void method2() {
            System.out.println("tentacles");
        }
        public String toString() {
            return "squid";
        }
    }
```

```
    // Class SeaCreatureApp
    public class SeaCreatureApp {
        public static void main (String []args){

            SeaCreature[] elements = {new Squid(), new
                Whale(), new SeaCreature(),new Mammal()};

            for (int i = 0; i < elements.length; i++) {
                System.out.println(elements[i]);
                elements[i].method1();
                elements[i].method2();
                System.out.println();
            }
        }
    }
```

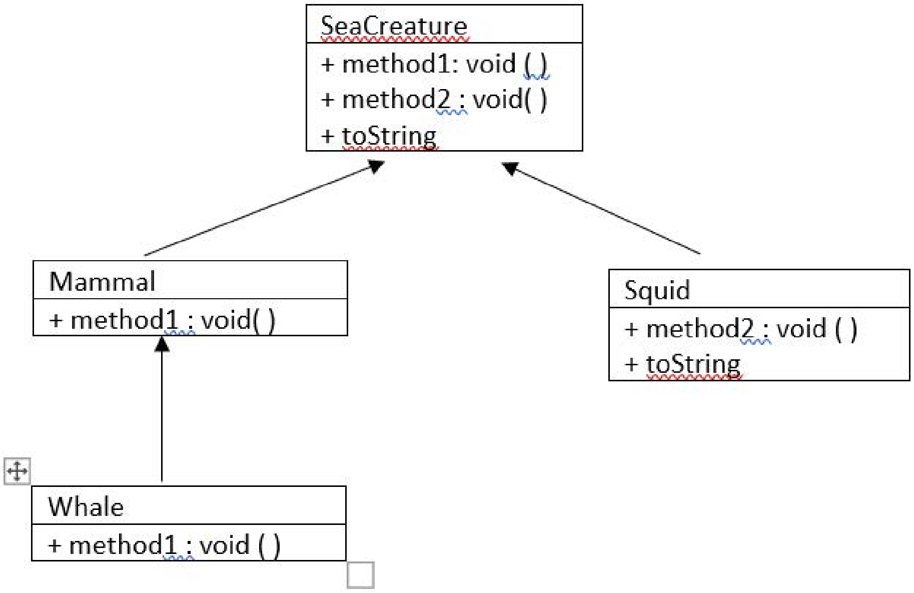
Answer:

```
squid              ocean dwelling
creature 1         creature 1
tentaclass         creature 2



BIG!               ocean dwelling
spout              warm blooded
creature 2         creature 2
```

b) Based on your observations, does the program uses an overloading OR overriding method? Explain your answer.

Program uses an overriding method.
Because the method holds the same name and parameters in the super class and its subclass

c) Draw a UML class diagram for java code program in question 2(a).



3. Identify **<u>NINE</u>** syntax errors in the following Java program segment and modify the program to remove the errors.

```
//Class Student
public class Student {
    private String name;
    private String matricNum;

    public Student(String n, String m) {
       name = n;
       matricNum = m;
    }
    public void displayDetails() {
       System.out.println("Name: " + name);
       System.out.println("Matric #: " + matricNum);
    }
 }
 //Class UnderGradStudent
 public class UnderGradStudent extends Student {
    private String program;
    public UnderGradStudent(String n, String m, String k);
     super(n,m);
     program = k;
    }
 }

 // class PostGradStudent
 public class PostGradStudent extends Student {
    private String supervisor;

 public PostGradStudent(String n, String m, String s)
   {
       super(n,m);
       s = supervisor;
    }

 public void displayDetails() {
       System.out.println("Name: "+name);
       System.out.println("Matric #: "+matricNum);
       System.out.println("Supervisor: "+supervisor);
    }
 }
```

Answer :

*** make a correction on code*

4. Please complete code based on the output given.

```
class Bird {
//a) the constructor class Bird and method Fly


}


public class BirdApp {

  public static void main(String[] args) {

         //b instantiate the Birds Object

         myBird.fly();
         myBird.fly(10000);
         myBird.fly("eagle", 10000);

        }
   }
```

Output :

```
The bird is flying.
The bird is flying 10000 feet high.
The eagle is flying 10000 feet high.
```

Answer a):

```
public class Bird {
        private String name;
        private int height;

        public void fly() {
                System.out.println("The bird is Flying.");
        }
        public void fly (int h) {
                this.height=h;
                System.out.println("The bird is flying " +height+ " feet high.");
        }
        public void fly(String n, int h) {
                this.name=n;
                this.height=h;
                System.out.println("The " +name+ " is flying " +height+ " feet high.");
        }
}
```

Answer b):

```
public static void main(String[]args) {
          Bird myBird = new Bird();
          myBird.fly();
          myBird.fly(10000);
          myBird.fly("eagle", 10000);
```

5. Why is following code showing compile error?

```
 public class AnimalApp{

        public static void main(String[] args){
        Animal[] a = {new Animal("Animal"),new Dog("Dog"),new Cat("Cat"),
                   new Lion("Lion")};
        for(int x=0 ; x<a.length; x++) {
              a[x].eat();}
}
}
class Animal{
        protected String animal_type;
        public Animal(String type) {
              animal_type=type;
        }
         void eat(){
            System.out.println(animal_type + " eating...");}
}
class Dog extends Animal{

         void eat(){System.out.println(animal_type + " eating
                  bone...");}
}
class Cat extends Animal{
        void eat(){System.out.println(animal_type + " eating
                 fish...");}
}
class Lion extends Animal{
        void eat(){System.out.println(animal_type + " eating
                  meat...");}
}
```

Answer a):

Because there is no constructor in the subclasses

b) Fix the error from the following code and display the output.

Answer b):

```
public class Dog extends Animal{
          public Dog(String type) {
                    super(type);
          }

public class Cat extends Animal{
          public Cat(String type) {
                    super(type);
          }

public class Lion extends Animal{
          public Lion(String type) {
                    super(type);
          }
```

Answer b (output):

```
Animal eating...
Dog eating bone
Cat eating fish...
Lion eating meat..
```

6. a) Based on the following code, can you find any problems? Explain your answer.

```java
class Loan {
        public double getRateOfInterest() {
                return 4.5;}
}
class Car extends Loan {
        public double getRateOfInterest() {
                return 3.6;}
        public String toString() {
                return "Car";}
}
class House extends Loan {
//c


}
class Land extends Loan {
        public double getRateOfInterest() {
                return 6.5;}
        public String toString() {
                return "Land";}
}
public class LoanApp {
       public static void main(String args[]) {
          Car[] loan= {new Car(), new House(), new Land()};
         for (int i=0;i<loan.length;i++) {
             System.out.println("Interest Rate for " +loan[i] + " is
             " +loan[i].getRateOfInterest() + "%");
         }

      }
}
```

Answer a):

In Line, the instance object created Car [ ] loan supposedly be Loan [ ] loan. This is because the class Loan is the super class so that the sub classes can inherit methods from the super class.

b) What is the output for the code after you fix the problem in a. Explain why it happen.

Answer b):

```
Interest Rate for Car is 3.6%
Interest Rate for House is 4.5%
Interest Rate for Land is 6.5%
```

c) Complete the class House so that the correct output can be generated.

Answer c):

```
class House extends Loan {
    public double getRateOf Interest( ) {
       return     ;}
    public String toString( ) {
       return "House";}
```

d) In your opinion, what is the purpose of toString Method in this code. Are we overload the method or override it? Share your idea.

Answer d):

Section B

The UML diagram and description is provided for each case. Your task is to write the program to implement the concept of inheritance and polymorphism for each case. You have to write a program of classes, which is one superclass, the others are subclasses and one application class.

Case 1 : UKM Members

The following UML diagram are show the various type of UKM Members. We have staff and students as a dominant members. Staff are divided into two which is Academic and non academic. Where as Student also have 2 category which is Undergraduate and post graduate. Define a staff, Student, Academic, Non-Academic, Undergraduate and Post-graduate Classes to performed the action. You also need to create an application class to show the difference of the output based on instruction given. Noted that **UkmMember** class are provided.

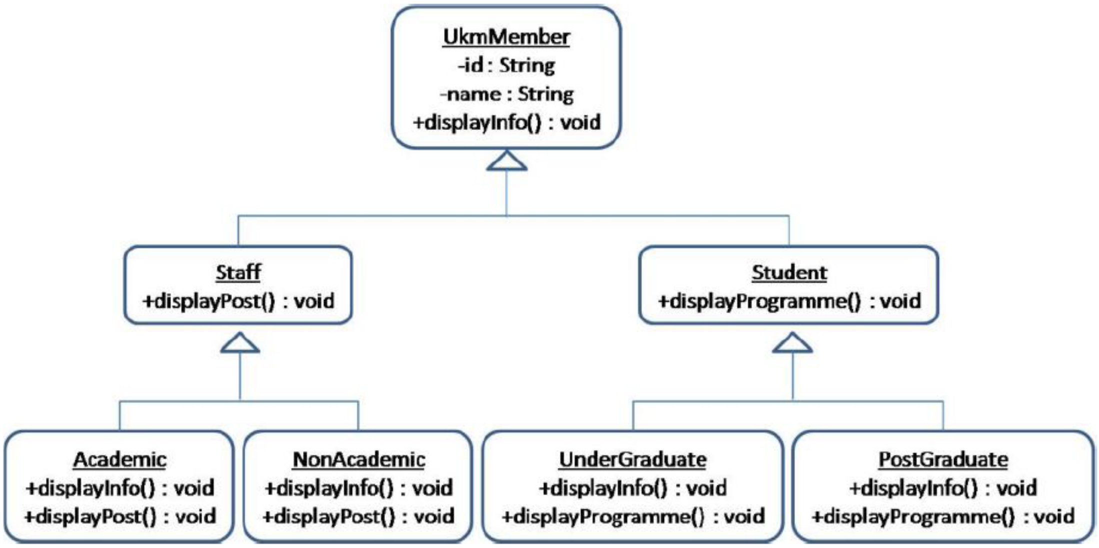


Figure 1 : Ukm Member

1. Define Staff, Academic, NonAcademic, Student, UnderGraduate and PostGraduate class
2. Define Application class that contains the main method, and inside the main method do the following:

a) Create an
- Academic staff object (id: K1487, name: Azzalea, post: lecturer),
- a NonAcademic staff object (id: K1294, name: Aiman, post: registrar),
- a UnderGraduate student object (id: P1263, name: Afreena, programme: Computer Science) and
- a PostGraduate student object (id: K1723, name: Mierza, programme: Object-Oriented Database Management System at UKM).

b) Create a suitable array to store all the objects created in (a).
c) Put the objects created in (a) into the array created in (b).
d) Use a for-loop to access each of the objects stored in the array created in (b), and:

- Execute displayInfo method of the object, then

- Execute displayPost method of the object, if the object is of staff of UKM, but if the object is of student of UKM, execute displayProgramme method of the object. Tip: Use if else-if and instanceof operator.

The output for the execution of application class should be as follows:

```
Azzalea (K1487) is an academic staff of UKM
Azzalea (K1487) is a lecturer at UKM
Aiman (K1294) is a non-academic staff of UKM
Aiman (K1294) is a registrar at UKM
Afreena (P1263) is an undergraduate student of UKM
Afreena (P1263) is studying Computer Science at UKM
Mierza (P1723) is a post-graduate student of UKM
Mierza (P1723) is doing research in Object-Oriented Database Management System at UKM
```

```java
//Class UkmMember : UkmMember.java

public class UkmMember {
	protected String id;
	protected String name;

	public UkmMember(String id, String name) {
		this.id = id;
		this.name = name;
	}

	public void displayInfo() {
		System.out.println(name +" ("+ id +")" +" is a member of UKM");
	}
}
```

Case 2 : Kuih oh Kuih

The MalayKuih consists of three types, and Figure 2 represent the UML diagram of MalayKuih (KuihBakar, Karipap and LepatPisang). Each kuih has different ways to cook and for certain type of kuih they have a filling. Write a program for class KuihBakar, Karipap, LepatPisang KaripapSardin and KaripapAyam. You also need to create Application class to test the class and method are created previously. Noted that MalayKuih class are provided.

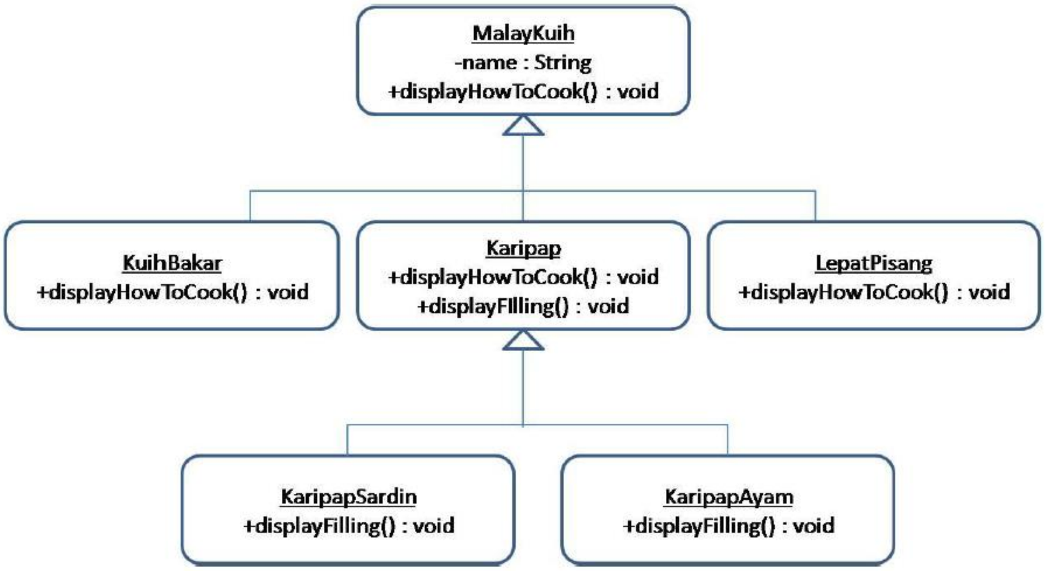


Figure 2: Malay Kuih UML

The output for the execution of Application class should be as follows:

```
Kuih bakar is cooked by baking
Karipap sardin is cooked by frying
Karipap sardin is filled with sardin
Karipap ayam is cooked by frying
Karipap ayam is filled with chicken
Lepat pisang is cooked by steaming
```

```java
//Class MalayKuih : MalayKuih.java

public class MalayKuih {
	protected String name;

	public MalayKuih(String name) {
		this.name = name;
	}
	public void displayHowToCook() {
		System.out.println("Depends on the type of kuih");
	}
}
```