INSTITUTE OF SOFTWARE ENGINEERING

GRADUATE DIPLOMA IN SOFTWARE ENGINEERING

Object Oriented Programming

ASSIGNMENT NO

02

NUMBER OF QUESTIONS: 37

NUMBER OF COMPLETED QUESTIONS: 35

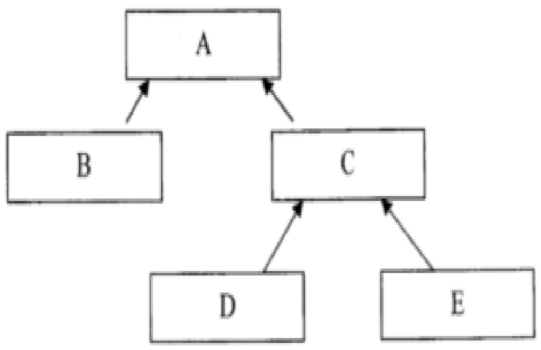
NUMBER OF REMAINING QUESTIONS: 02

STUDENT NAME: M.W.Maheeshi Naveendana Jayarathna

NIC: 200062000066

BATCH NO: GDSE 61

1.



A. Book

B. StoryBook

C. Dictionary

D. FantacyStoryBook

E. ChildrenStoryBook

```
class Tute2Test{
        public static void main(String args[]){
                Book book1 = new Book();
                StoryBook book2 = new StoryBook();
                Book book3 = new FantacyStoryBook();
                FantacyStoryBook book4 = new FantacyStoryBook();
                Book book5 = new ChildrenStoryBook();
                ChildrenStoryBook book6 = new ChildrenStoryBook();
                Dictionary book7 = new Dictionary();

                book1.setTitle("Movies");
                System.out.print("Title: "+book1.getTitle());

                book1.setAuthor("James");
                System.out.print("\tAuthor: "+book1.getAuthor());

                book1.setPages(335);
                System.out.print("\tPage Count: "+book1.getPages()+"\n");
```

```
		}



		book2.print("Butterflies","Tom");


		book3.setTitle("Little Mermaid");
		book3.setAuthor("Leo");
		System.out.println("Fancies > Title: "+book3.getTitle()+" By
"+book3.getAuthor());


		book4.reviews();


		book5.setTitle("Famous Five");
		book5.setAuthor("Enid Blyton");
		System.out.println("Children's Corner > Title: "+book5.getTitle()+" By
"+book5.getAuthor());


		book6.distribute();


		book7.readerCount();


	}
}

class Book{
	private String title;
	private String author;
	private int pages;



	public void setTitle(String title){
		this.title = title;
	}

	public String getTitle(){
		return title;
	}

	public void setAuthor(String author){
		this.author = author;
	}

	public String getAuthor(){
		return author;
	}

	public void setPages(int pages){
		this.pages = pages;
	}

	public int getPages(){
		return pages;
	}
}

class StoryBook extends Book{
	String title;
	String author;
	int starRate;

	public void print(String title, String author){
		this.title=title;
		this.author=author;
		System.out.println("Title: "+title);
		System.out.println("Author: "+author);
	}
}
```

```
class Dictionary extends Book{
        static int readers = 225;

        public void readerCount(){
                System.out.println("No of readers: "+readers);
        }
}

class FantacyStoryBook extends StoryBook{
        int starRate = 4;

        public void reviews(){
                System.out.println("Star Rate: "+ starRate);
        }
}

class ChildrenStoryBook extends StoryBook{
        String distribution = "World wide";

        public void distribute(){
                System.out.println("Distributions: "+ distribution);
        }
}
```

2.

- Allowing the methods and attributes (properties) of a certain class to be accessed by another class an in that class is called inheritance.
  Ex :- child and parents , all the properties of father are inherited by his son.

3.

- Inheritance allows developers to create subclasses that reuse code declared already in a superclass. Avoiding the duplication of common functionality between several classes by building a class inheritance hierarchy can save developers a considerable amount of time. Similarly, placing common functionality in a single superclass, rather than duplicating the code in multiple unrelated classes, helps prevent the same errors from appearing in multiple source-code files. If errors occur in the common functionality of the superclass, the software developer needs to modify only the superclass's.

4.

-
  ```
  class Demo{
       public static void main(String args[]){
       }
  }

  class Student{
   }

  class Shape{
   }

  class Loan{
  }

  class Employee{
  }

  class BankAccount{
  }

  class GraduateStudent extends Student{
  }

  class UndergraduateStudent extends Student{
  }

  class Circle extends Shape{
  }

  class Triangle extends Shape{
   }

  class Rectangle extends Shape{
  }

  class Sphere extends Shape{
  }
  ```

```
class Cube extends Shape{
}

class CarLoan extends Loan{
}

class HomeImprovementLoan extends Loan{
}

class MortgageLoan extends Loan{
 }

class EmployeeFaculty extends Employee{
 }

class Staff extends Employee{
 }

class CheckingAccount extends BankAccount{
 }

class SavingAccount extends BankAccount{
 }
```

5.

-
  ```
  class Demo{
        public static void main(String argos[]){
        }
  }

  class Shape{
  }

  class TwoDimensionalShape extends Shape{
  }

  class Circle extends TwoDimensionalShape{
  }

  class Square extends TwoDimensionalShape{
  }
  ```

```
        class Tringle extends TwoDimensionalShape{
        }

        class ThreeDimensionalShape extends Shape{
        }

        class Sphere extends ThreeDimensionalShape{
        }

        class Cube extends ThreeDimensionalShape{
        }

        class Terahedron extends ThreeDimensionalShape{
        }
```

6.

```
class Tute2Test{
        public static void main(String args[]){
                Student std1 = new Student();
                UnderGraduateStudent std2 = new UnderGraduateStudent();
                Student std3 = new Graduate();
                Graduate std4 = new Graduate();

                Sophomore std5 = new Sophomore();
                Junior std6 = new Junior();
                Senior std7 = new Senior();

                GraduateStudent std8 = new GraduateStudent();
                MastersStudent std9  = new MastersStudent();
                DoctralStudent std10 = new DoctralStudent();

                Graduate std11 = new GraduateStudent();
                Graduate std12 = new MastersStudent();
                Graduate std13 = new DoctralStudent();

                std1.setName("Kamal");
```

```
		std1.setRegId(152123);
		System.out.println("Name:	"+std1.getName()+"	Registration	Id:
"+std1.getRegId());

		std2.study();

		std3.setName("Kumudu");
		std3.setRegId(152103);
		System.out.println("Name:	"+std3.getName()+"	Registration	Id:
"+std3.getRegId());

		std4.throwParty();

		std5.throwFreshersParty();
		std6.doAssignment();
		std7.readyForExam();

		std8.searchJobs();
		std9.doMasters();
		std10.throwParty();

		std11.throwParty();
		std12.throwParty();
		std13.throwParty();

	}
}

class Student{
	private int regId;
	private String name;
```

```
        public void setRegId(int regId){
                this.regId = regId;
        }

        public int getRegId(){
                return regId;
        }

        public void setName(String name){
                this.name = name;
        }

        public String getName(){
                return name;
        }
}

class UnderGraduateStudent extends Student{
        public void study(){
                System.out.println("Assignment Due today");
        }
}

class Graduate extends Student{
        public void throwParty(){
                System.out.println("Done and Dusted");
        }
}

class Sophomore extends UnderGraduateStudent{
        public void throwFreshersParty(){
                System.out.println("Lets Enjoy");
        }
```

```
}

class Junior extends UnderGraduateStudent{
        public void doAssignment(){
                System.out.println("Assignment Due today");
        }
}

class Senior extends UnderGraduateStudent{
        public void readyForExam(){
                System.out.println("Exam starts on next week");
        }
}

class GraduateStudent extends Graduate{
        public void searchJobs(){
                System.out.println("Apply this");
        }
}

class MastersStudent extends Graduate{
        public void doMasters(){
                System.out.println("Study");
        }
}

class DoctralStudent extends Graduate{
        public void throwParty(){
                System.out.println("Congratulations");
        }
}
```

7.

```
class Quadrilateral{
        protected int x1,x2,x3,x4,y1,y2,y3,y4;
        protected void setCoordinate(int a,int b,int c,int d,int e,int f,int g,int h){
                x1=a;
                y1=b;
                x2=c;
                y2=d;
                x3=e;
                y3=f;
                x4=g;
                y4=h;
        }
}

class Square extends Quadrilateral{
        Square(int a,int b,int c,int d,int e,int f,int g,int h){
                setCoordinate(a,b,c,d,e,f,g,h);
        }
        int area(){
                int d=(int)Math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
        return d*d;
        }
}

class Rectangle extends Quadrilateral{
        Rectangle(int a,int b,int c,int d,int e,int f,int g,int h){
                setCoordinate(a,b,c,d,e,f,g,h);
        }
        int area(){
                int d1=(int)Math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
                int d2=(int)Math.sqrt((x1-x4)*(x1-x4)+(y1-y4)*(y1-y4));
                return d1*d2;
        }
}


class Trapezoid extends Quadrilateral{
        private int height;
        Trapezoid(int a,int b,int c,int d,int e,int f,int g,int h,int height){
                setCoordinate(a,b,c,d,e,f,g,h);
                this.height=height;
        }
        int area(){
                int d1=(int)Math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
                int d2=(int)Math.sqrt((x3-x4)*(x3-x4)+(y3-y4)*(y3-y4));
        return (int)((d1+d2)*height)/2;
        }
}

class Parallelogram extends Quadrilateral{
        private int height;
        Parallelogram(int a,int b,int c,int d,int e,int f,int g,int h,int height){
                setCoordinate(a,b,c,d,e,f,g,h);
                this.height=height;
        }
        int area(){
                int d1=(int)Math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
                return d1*height;
        }
}



class Tute2Test{
public static void main(String []args){
```

```
            Square sq=new Square(10,10,20,10,20,20,10,20);
            System.out.println("Area Of The Square is " + sq.area());
            Rectangle rec=new Rectangle(10,10,30,10,30,20,10,20);
            System.out.println("Area Of The Rectangle is " + rec.area());
            Parallelogram p=new Parallelogram(10,10,30,10,20,20,0,20,8);
            System.out.println("Area Of The Parallelogram is " + p.area());
            Trapezoid t=new Trapezoid(10,10,30,10,40,20,0,20,8);
            System.out.println("Area Of The Trapezoid is " + t.area());
            }
    }
```

8.

- CalculatorView class can extend the JFrame Class so the methods in JFrame Class can be used to create the CalculatorView Class because the properties in JFrame class inherit to the CalculatorClass from superclass JFrame. The setSize(), set Title(), setDefaultCloseOperation(), setVisible () methods can be accessible by the sub class. Because of that inheritance concept from oop gains the advantage of reusability of software.

9.

- A , B , C , D , E , F

10.

- Line 2 - printA() method cannot be accesible in Class B because printA method belongs to only class A.

  Line 4 - instance variable a is not declared in Class B and it belongs to Class A only. So int variable a cannot accible in Class B.

11.

- B and C

12.

- A , C , D

13.

- A , B , C , D

14.

- Compile error
  Constructor A in class A has passed parametrs int variable i
  So the Class A cannot be applied here. So the class B cannot extend Class A.

15.

- Line 9 - supper class object cannot assign to sub class type reference.
  Line 10 - Class C does not extends Class B so class C object cannot assign to class B references.
  Line 12 - upper class object cannot assign to sub class type reference.
  Line 13 - Class C does not extends Class B so class B object cannot assign to class C reference.
  Line 14 - Class C does not extends Class D so class D object cannot assign to class C reference.
  Line 15 - supper class object cannot assign to sub class type reference.
  Line 16 - supper class object cannot assign to sub class type reference.
  Line 17 - supper class object cannot assign to sub class type reference.

16.

- B

17.

- D

18.

- C , E

19.

- Compile error
  If the super class constructor has arguments we need to pass arguments to super in order to extend a class.

20.

- C - no issue x variable is a static varible belongs to super class and it can pass as argument to super constructor with parameters

  D - No issue y variable is a static variable belongs to sub class and subclass extends the super class and it can pass as argument to super constructor with parameters

}

E - No issue i variable belongs to sub constuctor and it can pass as argument to super constructor with parameters

F - 100 is an int value which can pass as argument to super constructor with parameters

21.

-
  ```
  class AnimalTest{
          public static void main(String args[]){
                  Animal animal2 = new Dog(4);
          }
  }


  class Animal{
          int countOfAnimal;


          public Animal(){
                  System.out.println("Constructor1");
          }


          public Animal(int countOfAnimal){
                  this.countOfAnimal = countOfAnimal;
                  System.out.println("Constructor2");
          }
  }


  class Dog extends Animal{
          public Dog(int countOfDogs){
                  super(countOfDogs);
                  System.out.println("Constructor in Dog");
          }
  }
  ```

  When we create a Dog object and passing 4 as argument to Dog constructor and assign it to Animal reference variable, it firstly call the the super class constructor (Animal constructor with int parameter countOfAnimal) then moves to Dog constructor.

22.

- Because variables in Java do not follow polymorphism and overriding is only applicable to methods but not to variables. And when an instance variable in a child has the same name as an instance variable in a parent class, then the instance variable is chosen from the reference type. In ava when we define a variable in child class with a name which we have already used to define a variable in the parent class, child class's variable hides parent classes variable even if their types are different. And this concept known as variable hiding.

23.

- F

24.

- F

25.

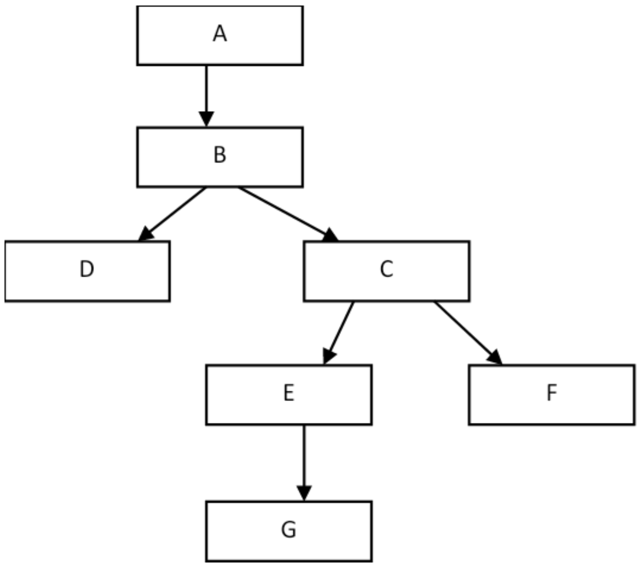
26.

- The subclass Rectangle extends Figure. Also the sub class Triangle class extend the class Figure. Also the subclass and the superclass both have method called area which performs method overriding. The sub class objects assigns to a super class reference variable figuref and figuret. When the overrided method called on respective reference variable it runs the method which belongs to the object. That is called run time polymorphism.

27.

- B

28.



29.

- D

30.

- A

31.

- B, C, D, E, F

32.

- D

33.

- A

34.

- B and C

35.

36.

37.

- D