



**JOMO KENYATTA UNIVERSITY
OF
AGRICULTURE AND TECHNOLOGY**

University Examinations

STAGE ONE SEMESTER THREE EXAMINATION FOR THE DEGREE
OF BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

ICS-2201 OBJECT ORIENTED PROGRAMMING-II

TIME : 2 HOURS

By : Mrs. Payal Doshi,

INSTRUCTIONS

1. Question ONE is compulsory.
2. Attempt question ONE and any other two.

Question One [Compulsory]

(a) List any five features of Java.

[5]

Answer :

1. Compiled and Interpreted
2. Platform Independent and Portable
3. Objected-Oriented
4. Distributed
5. Simple, Small and Familiar
6. Multithreaded and Interactive
7. High Performance

(b) What is garbage collection?

[2]

Answer:

Java handles deallocation of memory automatically. The technique that accomplishes this is called garbage collection.

(c) What is final?

[2]

Answer :

If you want to declare constant in java, variable is declared as final.

e.g. final double PI = 3.14 ;

It's a common coding convention to choose all uppercase identifiers for final variables.

(d) Explain the following terms.

[8]

1. Inheritance

- The mechanism of deriving a new class from an old one is called Inheritance. The old class is known as base class and the new one is called derived class or subclass.
- A class can also inherit properties from more than one class or from more than one level.
- A derived class with only one base class is called single inheritance and one with several base classes called multiple inheritances.
- The mechanism of deriving a class from another 'derived class' is known as multi-level inheritance.
- The traits of one class may be inherited by more than one class. The process is known as hierarchical approach.

2. Encapsulation

- The wrapping up of data and functions in to a single unit (class) is known as encapsulation.
- The data is not accessible to the outside world, and only those functions, which are wrapped in the class, can access it.
- These functions provide interface between the object's data and the program. This insulation of the data from direct access by the program is called data hiding or information hiding.

3. Polymorphism

- Polymorphism means the ability to take more than one form. An operation may exhibit different behavior in different instances. The behavior depends upon the types of data used in the operation.
- The single function name can be used to handle different member and different types of arguments is known as function overloading.
- Polymorphism allows objects having different internal structures to share the same external interface.

4. Exception

- An exception is an object that is generated at run time to describe a problem occurred during the execution of a program.
- Some causes for an exception are integer division by zero, array index out of bound, incorrect number format etc.
- The Java allows you to handle exceptions that occur during execution of a program. This is done by using following syntax

```
try {  
    // try block  
}  
catch (Exception type1 param1) {  
    //Exception block  
}
```

(e) Differentiate between method overriding and method overloading with example.

[4]

- When a method in subclass has the same name and type signature as method in its superclass, then the method is said to be override the method in superclass.
- When an overridden method called from subclass it will always refer subclass's method.
- Overridden methods allow java to support run time polymorphism.

```
class A {
    int i, j;
    A(int a, int b) {
        i = a;
        j = b;
    }

    void show() {
        System.out.println("i and j are" + i + " " + j);
    }
}
class B extends A {
    int k;

    B(int a, int b, int c) {
        super(a,b);
        k = c;
    }

    void show() {
        System.out.println("k = " + k);
    }
}
class Override {
    public static void main(String s[]) {
        B b = new B(10,20,30);
        b.show();
    }
}
```

- When a method in the class has the same name and arguments are different, then the method is said to be overloading.

```
class A {
    void show() {
        System.out.println("show function");
    }
    void show(int x) {
        System.out.println("x = " + x);
    }
}
```

```

class Overloading {
    public static void main(String s[]) {
        A a = new A();
        a.show();
        a.show(100);
    }
}

```

(f) Explain the following. [2]
public static void main(String s[])

Answer :

- public : it is an access specifier that declares the main method as unprotected. It is similar to the C++ public modifier.
- static : this declares that method is belongs to the entire class not to the part of any objects of that class.
- void : this states that main method does not return any value.

(g) What is an applet? [1]

Answer :

- An applet is a program that can be referenced by the HTML source code of a web page. It is dynamically downloaded from the web server to a browser.

(h) How will you declare an array in Java? [2]

Array_var = new datatype [size] ;
 e.g. x = new int[5];

(i) List eight primitive datatypes of Java. [4]

Answer:

byte, short, int, long, float, double char, boolean

Question Two

(a) Define the following terms with example. [6]

1. Class

- The entire set of data and code of an object can be made a user defined data type with the help of a class.
- Objects are variables of class.
- Once a class has been defined, we can create number of objects belonging to that class.
- A class is a collection of objects of similar type. E.g. mango, apple, orange are members of Fruit class.

2. Method

Methods are functions that are used to perform operation on object. It may use the object's properties to perform these operations.

3. Object

- Objects are basic run-time entities in an Object-Oriented system. They may represent a person, a place, a bank account, and a table of data or may it that program has to handle.
- For example if customer and account are two objects and then customer object may send a message to the account object requesting for the bank balance.
- Each object contains data and code to manipulate the data.
- Objects can interact without having to know details of each other's data or code.

Object : STUDENT
DATA Name DOB Marks
Functions: Total Average Display

(b) Write a program that accepts the radius of a circle as its command-line argument and displays the area of this circle. [5]

Answer :

```
class Area {
    public static void main(String s[])
    {
        double r = Double.parseDouble(s[0]);
        double area = 3.14 * r * r;
        System.out.println("Area of a circle is " + area);
    }
}
```

(c) Write an application that declares a class named Person. It should have instance variables to record name, age and salary. These should be of types String, integer and float. Create a Person object set and display its instance variables. [7]

Answer:

```
class Person {
    String name;
    int age;
    float salary;
}
class PersonExample {
    public static void main (String s[])
    {
```

```

        Person p = new Person();
        p.name= "John";
        p.age = 21;
        p.salary = 10000.50f;
        System.out.println("name is "+p.name);
        System.out.println("age is "+p.age);
        System.out.println("salary is "+p.salary);
    }
}

```

(d) Find out errors or output from the following segment.

[2]

```

class Test
{
    public static void main(String []s)
    {
        System.out.println("Hello "+"Welcome to B.Sc.");
    }
}

```

Answer:

Hello Welcome to B.Sc.

Question Three

(a) Explain the following keywords with example.

[9]

1. **extends**

When you inherit, you say "This new class is like that old class." You state this in code by giving the name of the class as usual, but before the opening brace of the class body, put the keyword **extends** followed by the name of the *base class*. When you do this, you automatically get all the data members and methods in the base class. With the help of extends keyword you can inherit base class to the derive class.

e.g.

```

class A {
    double d, h, w;

    A() {
        d = 10; h = 20; w = 30;
    }

    double volume() {
        return d * w * h;
    }
}

class B extends A {
    double wt;
    B(double wt) {

```

```

        this.wt = wt;
        System.out.println("Weight = " + wt);
    }
}
class Inherit {
    public static void main(String s[]) {

        B b = new B(10);
        double vol;
        vol = b.volume();
        System.out.println("Volume == " + vol);

    }
}

```

2. abstract

If we want to overridden certain methods by subclass we have to use abstract modifier. These methods are sometimes referred as to as subclasses responsibility because they have no implementation. To declare an abstract method, we use following form:
 abstract type name (parameter_list)

e.g.

```

abstract class A {
    abstract void display() ;
    void show() {
        System.out.println("This is show method");
    }
}
class B extends A {
    void display() {
        System.out.println("in B override display");
    }
}
class AbstractDemo {
    public static void main(String s[]) {
        B b = new B() ;
        b.display() ;
        b.show() ;
    }
}

```

3. super

A subclass can call a constructor defined by its superclass by use of super keyword.
 super(parameter_list)
 super must be the first statement executed inside a subclass constructor.

e.g.

```

class A {

```

```

double d, h, w;
A(double d, double h, double w) {
    this.d = d;
    this.h = h;
    this.w = w;
}
double volume() {
    return d * w * h;
}
}
class B extends A {
    double wt ;
    B(double wt) {
        super(2,3,4);
        this.wt = wt;
        System.out.println("Weight = " + wt);
    }
}
class Inherit {
    public static void main(String s[]) {
        B b = new B(10);
        double vol;
        vol = b.volume();
        System.out.println("Volume == " + vol);
    }
}
}

```

(b) Assume that the test results of a batch of students are stored in three different classes. Create a class student stores the roll number, a class test inherits class student and stores the marks obtained in three subjects, and a class result inherits class tests contains the total marks and percentage obtain in the test. Include necessary methods to achieve the following tasks :

1. Calculate Total and Percentage
2. Display roll number, marks, for three subjects, total and percentage. [11]

Answer :

```

class Student {
    int rollno;
    void getno(int rollno) {
        this.rollno = rollno;
    }
}
class Tests extends Student {
    int mark1, mark2, mark3, sum;
    float per;
    void getdata(int mark1, int mark2, int mark3) {

```



```

        this.mark1=mark1;
        this.mark2=mark2;
        this.mark3=mark3;
    }
}
class Result extends Tests {
    void calc()    {
        sum = (mark1+ mark2 + mark3);
        per =sum /3;
    }
    void display()    {
        System.out.println("Roll no is " + rollno);
        System.out.println("mark1 is " + mark1);
        System.out.println("mark2 is " + mark2);
        System.out.println("mark3 is " + mark3);
        System.out.println("Total is " + sum);
        System.out.println("Per is " + per);
    }
}
}
class Stud {
    public static void main(String s[])    {
        Result r = new Result();
        r.getno(10);
        r.getdata(45,56,67);
        r.calc();
        r.display();
    }
}

```

Question Four

(a) What is an interface? What is the default modifier for an interface? [4]

Answer :

- An interface is a kind of class, which contain methods and variables.
- The difference is that interface defines only abstract methods and final fields.
- Variables are declared as follows :
Static final type variable_name = value;
- Methods are declared as follows :
return_type methodname (parameter_list);
- default modifier of an interface is static final.

(b) Differentiate between throw and throws. [2]

Answer :

Throw

- The general form of throw is :
- throw Throwableinstance ;

- Here, Throwable instance must be an object of type Throwable or a subclass of Throwable.
- There are two ways for obtaining Throwable object :
 - Using a parameter into a catch clause.
 - Creating with the new operator.
- The flow of execution stops immediately after the throw statement; any subsequent statements are not executed.

Throws

- If method is capable of causing an exception that is not handling, it must specify this behavior so that callers of that method can guard themselves against exception.
- We can do this by including throws clause in the method's declaration.
- A throws clause lists the types of exceptions that method might throw.
- All other exceptions that method throws must be declared in the throws clause, otherwise compilation error will generate.

(c) Write a program that accepts two strings as its arguments. It converts these to double values and display the sum. The program generates an error message if the format is invalid or if there are no arguments.

[10]

Answer :

```
class Excep {
    public static void main(String s[]) {
        try {
            double no1 = Double.parseDouble(s[0]);
            double no2 = Double.parseDouble(s[1]);
            double sum;
            sum = no1 + no2;
        }
        catch (NumberFormatException e)
        {
            System.out.println("Invalid Format "+ e);
        }
        catch(ArrayIndexOutOfBoundsException e1){
            System.out.println("pass argumetns "+ e1);
        }
    }
}
```

(d) List any four types of Exceptions.

[4]

1. NumberFormatException
2. ArrayIndexOutOfBoundsException
3. NoClassDefFoundException
4. NullPointerException

- Here, Throwable instance must be an object of type Throwable or a subclass of Throwable.
- There are two ways for obtaining Throwable object :
 - Using a parameter into a catch clause.
 - Creating with the new operator.
- The flow of execution stops immediately after the throw statement; any subsequent statements are not executed.

Throws

- If method is capable of causing an exception that is not handling, it must specify this behavior so that callers of that method can guard themselves against exception.
- We can do this by including throws clause in the method's declaration.
- A throws clause lists the types of exceptions that method might throw.
- All other exceptions that method throws must be declared in the throws clause, otherwise compilation error will generate.

(c) Write a program that accepts two strings as its arguments. It converts these to double values and display the sum. The program generates an error message if the format is invalid or if there are no arguments.

[10]

Answer :

```
class Excep {
    public static void main(String s[]) {
        try {
            double no1 = Double.parseDouble(s[0]);
            double no2 = Double.parseDouble(s[1]);
            double sum;
            sum = no1 + no2;
        }
        catch (NumberFormatException e)
        {
            System.out.println("Invalid Format "+ e);
        }
        catch(ArrayIndexOutOfBoundsException e1){
            System.out.println("pass argumetns "+ e1);
        }
    }
}
```

(d) List any four types of Exceptions.

[4]

1. NumberFormatException
2. ArrayIndexOutOfBoundsException
3. NoClassDefFoundException
4. NullPointerException

Question Five

(a) Differentiate between Java Applet and Application. [4]

(b) Write an applet to display "Hello World" in the center of the applet. [8]

Answer:

```
import java.applet.*;
import java.awt.*;
/* <Applet code="AppletString" width=300 height=300>
</applet>*/
```

```
public class AppletString extends Applet{
    public void paint(Graphics g) {
        setBackground(Color.blue);
        setForeground(Color.yellow);
        Dimension d = getSize();
        int x = d.width / 2;
        int y = d.height / 2;
        g.drawString("Hello World",x,y);
    }
}
```

(c) Explain two ways of executing an applet. [4]

Answer:

5. In Internet explorer – create HTML file and use <applet>.
6. In java program use <applet> with comments.

(d) Explain the life cycle of an applet. [4]

Answer :

- ⇒ void init()
It calls only when the applet begins execution.
- ⇒ void start()
It is executed after the init() method completes execution. This method is called by appletviewer or web browser to resume the execution.
- ⇒ void stop()
It is called by the appletviewer or web browser to suspend the execution of an applet. Therefore, the start() and stop() methods may be called multiple times during the life cycle of the applet.
- ⇒ void destroy()
It is called by the appletviewer or web browser before the applet is terminated. The stop() method invoked before destroy().