CS3391 – OBJECT ORIENTED PROGRAMMING

ANNAUNIVERSITY QUESTION PAPER – 2 MARKS

UNIT – I

1. **List the features of Object Oriented Programming.**
   i. Abstraction
   ii. Object and Classes
   iii. Encapsulation
   iv. Inheritance
   v. Polymorphism

2. **Give an example for defining constant in Java.**

We can use *final* keyword to declare constant in Java.

```
public class ConstantExample
{
final int A=10; //A is a constant
}
```

3. **What is byte code?**

Javac – Compiler converts source code to byte code

Java – Interpreter converts Byte code to Machine code

Byte code is executed by the Java Virtual Machine (JVM) on any device that supports Java.

4. **Write the general form of the for-each version of the for statement.**

```
for (Type variable : collection/array)
{
   // code to execute
}
```

5. **Java language is Platform independent. Justify your answer.**

Java language is Platform independent. Java compiler creates platform-independent byte code, which runs on any device.

6. **Difference between method and constructor.**

CONSTRUCTOR	METHODS
Executed when object created	Executed when we explicitly call it
Constructor name will be the same as Class name	Method name will not be the same as Class name
Should have return type	Should not have return type
Will get executed only once per object	Will get executed n number of times

7. **What is an array? How multidimensional arrays are implemented in Java?**

Array is a collection of elements of the same data type stored in contiguous memory locations.

```
int[][] num = {  {1, 2, 3},  {4, 5, 6},  {7, 8, 9}  };
```

8. **Name the access modifiers in Java.**

Access modifiers controls access to data fields, methods and classes.

The types are,

1. Public
2. Private
3. Protected
4. Default modifier

UNIT – II

1. Mention the use of constructor overloading.

- **Flexibility**
- **Code reusability**
- **Readability**
- **Maintainability**
- **Polymorphic behavior**
- **Resource management**

2. Illustrate method overriding with an example.

Same method name with same arguments

```
class ABCD // Super Class
{
int dept=5;
   void display( )
   {    System.out.println("Number of department="+dept);    }
}
class IT extends ABCD // Sub Class
{
int years=2;
   void display( ) // Method Overriding
   {    super.display( );System.out.println("Number of years="+years);    }
}
```

3. Differentiate method overloading and overriding.

Method Overloading	Method Overriding
1.Multiple methods of same name in single class.	1. Multiple methods of same name in different class.
2.No need of inheritance	2. Inheritance is used
3.All methods have different arguments.	3. All methods have same arguments.
4.It's a compile time polymorphism.	4. It's a run time polymorphism.
5.No special keyword used.	5. Virtual & override keywords.

4. **Can we access parent class variables in child class by using super keyword?**

   Yes. We can access parent class variable in child class by using super keyword.

```
class Parent
{
   int a = 10;
}
class Child extends Parent
{
   void print()
    {
     System.out.println("Value:" + super.a);
    }
  }
```

5. **What is the use of final keyword?**

   The final keyword is used to restrict changes to variables, methods, or classes

6. **How dynamic method resolution is achieved in Java?**

   Dynamic method resolution, also known as runtime polymorphism.

   Mechanism:

   - **Inheritance**
   - **Method Overriding**
   - **Up casting**
   - **Runtime Decision**

7. **Define inheritance.**

Deriving new class from existing class

Existing class – Parent Class or Super Class

New class – Child Class or Sub Class

extends – keyword used derive new class

super – keyword used to access the properties of super class

8. **How can a subclass can call a constructor defined by its super class?**

We can call super class constructor by using super keyword.

```
class ABCD // Super Class
{
int dept;
ABCD( )
{
dept=5;
}
}
class IT extends ABCD // Sub Class
{
int years;
IT( )
{
super( );
years=3;
}
}
```

UNIT – III

1. Outline the difference between unchecked exception and checked exception.

S.No.	Checked Exception	Unchecked Exception
1	Checked exceptions happen at compile time	Unchecked exceptions happen at runtime
2	Checked by the compiler	Not checked by the compiler.
3	Counted as a sub-class of the class	Not included in the exception class
4	JVM requires the exception to be caught or handled	JVM does not need the exception to be caught or handled.

2. Name the method used by Java for inter process communication to avoid polling.

wait( )
notify( )
notifyAll( )

3. Define arithmetic exception with example.

An arithmetic exception is an error that occurs when a mathematical operation results in an undefined or illegal result.

EX: Division by zero - int a = 10 / 0

4. Name the two ways to create a thread in Java.

i) By extending Thread class
ii) By implementing Runnable interface

5. Why to handle exception?

- Helps to identify errors
- Provides feedback
- Avoids unexpected behavior
- Completes program execution
- Helps to write cleaner code

6. **What is thread priority? How it can be set for thread?**

   Thread priority is an integer value that indicates the order of thread execution. We can call setPriority( ) method with an integer value to set priority.

7. **List the values associated with the parameters of setpriority( ) method of Thread class.**

Thread priority	Value	Description
Minimum	1	Less urgent, typically used for background tasks
Normal	5	Default priority, threads compete fairly for resources
Maximum	10	More urgent, often used for critical tasks

8. **Define deadlock.**

   In Java, a deadlock occurs when two or more threads are blocked forever, each waiting for the other to release a resource that it holds.

UNIT – IV

1. **What are streams?**

   streams are a functional programming construct to process collections of data in a declarative and efficient manner.

2. **Why parameterized streams are important?**

   Parameterized streams, also known as generic streams, are important in Java for the following reasons:

   - Type Safety
   - Code Reusability
   - Readability & Maintainability
   - Improved Performance

3. **What is Thread pool?**

   A thread pool is a managed collection of worker threads that can be reused to execute multiple tasks.

4. **When a class must me declared as abstract?**

   A class must be abstract if it contains an abstract method. An abstract method doesn't have a body.

5. **Write any four methods associated with basic string class.**

   - length( )
   - concat( )
   - replace( )
   - charAt( )

6. **What is a string buffer class?**

   StringBuffer class is used to create mutable (modifiable) string.

   The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.

7. **State the purpose of Valueof( ) method in string class.**

   The valueOf() method in the String class is used to return the primitive value of a String object.

8. **List any two methods available in Data Output Interface.**

   - void writeByte(int a)
   - void writeBytes(String s)
   - void writeChar(char c)
   - void writeChars(String s)

UNIT – V

1. **What is JavaFX?**

   JavaFX is a Java library and framework that allows developers to create applications for desktop, web, and mobile devices

2. **Write a note on HBox and VBox.**

   The HBox component is used to create a horizontally oriented box. Each component placed in the HBox will be placed horizontally in a row.

   The VBox component is used to create a vertically oriented box. Added components will be placed underneath each other in a column.

3. **What is a Layout manager and what are different types of Layout managers available in java AWT?**

   A layout manager in Java is an object that controls the position and size of components within a container.

   - Border Layout
   - Flow Layout
   - Grid Layout
   - GridBag Layout

4. **Differentiate HBox and VBox.**

HBOX	VBOX
“H” stands for horizontal	"V" stands for vertical
Arranges elements horizontally in a row	Arranges elements vertically in a column

5. **Why JavaFX is preferred for building internet applications?**

   Because it has many features that make it a good choice for developers:

   - Rich user interface
   - Easy to learn
   - Advanced features
   - Customization
   - Cross-platform compatibility

6. **Write the constructors of HBox class.**

   - public HBox()
   - public HBox(double spacing)
   - public HBox(double spacing, Node... children)
   - public HBox(Node... children)

7. **What is the use of adapter class?**

   An adapter class in Java is a design pattern used to allow two incompatible interfaces to work together. It acts as a bridge between two classes and enabling them to communicate.

8. **List any two forms of CheckBoxMenuItem constructor.**

   - CheckboxMenuItem()
   - CheckboxMenuItem(String label)
   - CheckboxMenuItem(String label, boolean state)