Name: QSmilin Jasper Reg.no: 20505015 Sub.Code: 35243

PART – A

1. What is mean by data encapsulation?

Ans:-

*The binding or wrapping code (methods) and data together into a single unit is known as encapsulation.

*The data is not accessible to the outside world.

2. What is array processing?.

Ans:-

*We know that array is a group of data of same data type, so to use the array elements we use any one of the looping statements. Normally for loop is used.

3. What is the use of this keyword?

Ans:-

*Sometimes a method will need to refer to the object that invoked it.

*'this' keyword is defined for the purpose.

*It can be used inside any method to refer to the current object. this always refers to the object on which the method was invoked.

* 'this' can be used anywhere, where a reference to an object of the current class type is permitted.

Date: 24.05.2021

Signature:
Smilin

Name: QSmilin Jasper Reg.no: 2050505015 Sub.Code: 35243

4. What are applets?

Ans:-

*Applets are small Java programs developed for Internet applications.

* An applet located in distant computer can be downloaded via Internet and executed on a local computer using Java enabled browser.

* The applets are restricted to access the local file system and the resources.

* The Java applets can also be executed in the command line using appletviewer, a JDK tool.

7. Define file.

Ans:-

* A file is a collection of related records. A record is the collection of fields. A field is a group of characters.

PART – B

1. What is mean by identifier?

Ans:-

* Identifiers are used for class names, method names, and variable names.

* An identifier may be any descriptive sequence of uppercase and lowercase letters, numbers, or the underscore and dollar-sign characters.

Date: 24.05.2021

Signature:
Smilin

* They must not begin with a number.

* Java is casesensitive, so VALUE is a different identifier than Value.

* Some examples of valid identifiers are:-

- AvgTemp
- count
- a4
- this_is_ok

4. List the different system packages?

Ans:-

The frequently used system packages are:-

1) java.lang:-

This is the default package that is automatically imported and it contains classes that support the java language including primitive types, strings, Math functions, Threads and exceptions.

2) java.util:-

This contains classes for utilities such as vectors, hash tables, random nos., date etc.

3) java.io:-

This contains classes for supporting input and output operations.

Date: 24.05.2021

Signature:
Smilin

4) java.awt:-

This contains classes for implementing GUI elements such as windows, buttons, lists, menus.

5) java.net:-

This contains classes for networking with local computers as well as with the internet servers. 6) java.applet This contains classes for creating and implementing the applets.

5. What are the types of applets?

Ans:-

Local Applets:-

* Local applet They are developed locally and stored in a local system.

* Internet connection is not required. When a web page is trying to find a local applet, it simply searches the directories in the local system and locates and loads the specified applet.

Remote Applets:-

* A remote applet is the one developed by the remote user and stored on a remote computer connected to the Internet.

* A remote applet can be downloaded on to our system through the Internet and executed.

* To locate and load a remote applet, the applets address should be known. This address is known as Uniform Resource Locator (URL) and must be specified in the

Date: 24.05.2021

Signature:
Smilin

applets HTML document as the value of the CODEBASE attribute.

Eg;- CODEBASE = http://www.netserve.com/applets

* If the applet is local, CODEBASE may be absent or may specify a local directory.

7. Write any methods of OutputStream class and state their use.

Ans:-

* OutputStream class is an abstract class.

* It is the superclass of all classes representing an output stream of bytes. The methods in this class are given below:-

Method	Use
void write(int)	It is used to write a byte to the current output stream
void write(byte[ ] b)	It is used to write an array of byte to the current output stream
void write(byte[ ] b ,int m, int n)	It writes n bytes from buffer array of data from the mth position.
void flush()	It flushes the current output stream.
void close()	It closes the current output stream

Date: 24.05.2021

Signature:
Smilin

Name: QSmilin Jasper Reg.no: 20505015 Sub.Code: 35243

8. Write any methods of Reader class and state their use.

Ans:-

* Reader class is an abstract class.
* It is the superclass of all classes representing an input stream of characters. The methods in this class are given below:-

Method	Use
int available()	It returns the number of characters that can be read from the current input stream.
int available()	It reads the next character of data from the input stream. It returns -1 at the end of file.
int read(char [] b)	It reads some number of characters from the input stream and stores them into the buffer array b.
int read(char[] b, int m, int n)	It reads up to n characters of data from the mth position of the input stream into an array of bytes.
void reset().	It moves the pointer to the beginning of the input stream
long skip(long n)	It skips n characters of data from the input stream
void close()	It closes the current input stream.

Date: 24.05.2021

Signature:
Smilin

Name: QSmilin Jasper Reg.no: 2050501S Sub.Code: 35243

PART – C

1. a) Explain about the history of Java.

Ans:-

History Of Java:-

* Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991.

* It took 18 months to develop the first working version. This language was initially called "Oak," but was renamed "Java" in 1995.

* Between the initial implementation of Oak in the fall of 1992 and the public announcement of Java in the spring of 1995, many more people contributed to the design and evolution of the language.

* Bill Joy, Arthur van Hoff, Jonathan Payne, Frank Yellin, and Tim Lindholm were key contributors to the maturing of the original prototype.

* The primary motivation was the need for a platform-independent (that is, architecture-neutral) language that could be used to create software to be embedded in various consumer electronic devices, such as microwave ovens and remote controls.

* So many different types of CPUs are used as controllers. The problem is that compilers are expensive and time-consuming to create for different types of CPU.

* An easier—and more cost-efficient—solution was needed.

* In an attempt to find such a solution, Gosling and others began work on a portable,

Date: 24.05.2021

Signature:
Smilin

platform-independent language that could be used to produce code that would run on a variety of CPUs under differing environments.

* This effort ultimately led to the creation of Java. About the time that the details of Java were being worked out, a second, and ultimately more important, factor was emerging that would play a crucial role in the future of Java.

* This second force was, of course, the World Wide Web. Java might have remained a useful but obscure language for programming consumer electronics.

* However, with the emergence of the World Wide Web, Java was propelled to the forefront of computer language design, because the Web, too, demanded portable programs.

2. a) Explain the switch statement with example.

Ans:-

* The switch statement is Java's multi-way branch statement.

* It provides an easy way to dispatch execution to different parts of your code based on the value of an expression.

* As such, it often provides a better alternative than a large series of if-else-if statements.

Here is the general form of a switch statement:-

```
switch (expression) {
case value1:
```

Date: 24.05.2021

Signature:
Smilin

```
// statement sequence
break; case value2:
// statement sequence
break;
...
case valueN:
// statement sequence
break;
default:
// default statement sequence
}
```

* The expression must be of type byte, short, int, or char; each of the values specified in the case statements must be of a type compatible with the expression.

* Each case value must be a unique literal (that is, it must be a constant, not a variable).

* Duplicate case values are not allowed.

The switch statement works like this:-

* The value of the expression is compared with each of the literal values in the case statements.

* If a match is found, the code sequence following that case statement is executed.

* If none of the constants matches the value of the 29 expression, then the default statement is executed. However, the default statement is optional.

Date: 24.05.2021

Signature:
Smilin

* If no case matches and no default is present, then no further action is taken.

* The break statement is used inside the switch to terminate a statement sequence.

* When a break statement is encountered, execution branches to the first line of code that follows the entire switch statement.

* This has the effect of "jumping out" of the switch.

Here is a simple example that uses a switch statement:-

```
// A simple example of the switch.
class SampleSwitch { public static void main(String args[]) {
for(int i=0; i<6; i++)
switch(i) {
case 0:
system.out.println("i is zero.");
break;
case 1:
system.out.println("i is one.");
break;
case 2:
system.out.println("i is two.");
break;
case 3:
system.out.println("i is three.");
```

Date: 24.05.2021

Signature:
Smilin

```
break;
default:
system.out.println("i is greater than 3.");
}
}
}
```

The output produced by this program is shown here:

```
i is zero.
i is one.
i is two.
i is three.
i is greater than 3.
i is greater than 3.
```

3) a) Explain constructors with example.

Ans:-

Constructors: -

A constructor is a special method used to initialize an object at the time of creation.

Its general form is:-

```
Constructor name (arguments)
{
Initialization statements;
}
```

Date: 24.05.2021

Signature: Smilin

where constructor name is same as that of the class name. Constructors return instance of a class and hence they do not specify a return type. A constructor may or may not have arguments.

Eg;-

1) constructor without arguments:

```
Rectangle ( )
{
length= 15;
width = 10;
}
```

2) Constructor with arguments:-

```
Rectangle (int x, int y)
{
length =x;
width = y;
}
```

eg;-

```
class Rectangle
{
int length, width;
Rectangle (int x, int y)
{
```

Date: 24.05.2021

Signature:
Smilin

```
length =x;
width = y;
}
int rectArea()
{
return (length * width);
}
}
class RectArea
{
public static void main (String args[])
{
Rectangle r1 = new Rectangle(15, 20);
int area1 = r1.rectArea();
System.out.println("Area of rectangle =" + area1);
}
}
```

4) b) Explain key events with example?

Ans:-

* When an event is generated through the key board, the keyHandler class responds.

* We can generate three types of key events. They are:

Date: 24.05.2021

Signature:
Smilin

1. Pressing a key (holding a key pressed)
2. Releasing a pressed key
3. Typing a key

The KeyListener interface has three methods corresponding to the above events.

They are:

- keyPressed (KeyEvent e)
- keyReleased (KeyEvent e)
- keyTyped (KeyEvent e)

* All the methods take an object of KeyEvent class as the parameter.

* The KeyEvent class has a method getKeyChar( ) which returns the character of the key from which the event is generated.

Eg;-

if e is a KeyEvent, then

Char c;

c = e.getKeyChar ( );

Returns the character in c.

Eg;-

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
```

Date: 24.05.2021

Signature:
Smilin

Name: QSmilin Jasper Reg.no: 20505015 Sub.Code: 35243

```
public class KeyEventDemo extends Applet
{
String msg = "";
public void init ( )
{
addKeyListener (new KeyManager ( ));
requestFocus ( );
}
public void paint (Graphics g)
{
font f = new Font ("Ariel", Font Bold, 30);
g.setFont (f);
g.drawString (msg, 50, 50);
}
class KeyHandler implements KeyListener
{
public void KeyPressed (KeyEvent e)
{
char c = e.getKeyChar ( );
switch ( )
{
case 'R':
case 'r' : setBackground (new Color (255,0,0));
msg = "Red";
break;
```

Date: 24.05.2021

Signature:
Smilin

```
case 'G':
case 'g': setBackground (new Color (0,255,0));
msg = "green";
break;
case 'B':
    case 'b': setBackground (new Color (0,0,255));
msg = "blue"
break;
}
repaint ();
}
public void keyReleased (KeyEvent e)
{
}
public void keyTyped (KeyEvent e)
{
}
}
}
```

5) a) How you define and run a thread using Runnable interface?. Give example.

Ans:-

Defining & Running Thread:-

The thread can be defined by extending the Thread class.

Date: 24.05.2021

Signature:
Smilin

Name: QSmilin Jasper Reg.no: 20505015 Sub.Code: 35243

1. Define subclasses by extending the Thread class

```
class classname extends Thread
{

}
```

2. Override the run() method in all subclasses.

```
public void run ()
{
}
```

3. Create thread object in the main() method

```
classname object = new classname();
```

4. Call the start method using the thread object.

```
Object.start();
```

Example

```
class Even extends Thread
{
public void run()
{
for(int i=0; i<=10; i=i+2)
```

Date: 24.05.2021

Signature:
Smilin

```
system.out.println("Even number: "+i);
}
}
class Odd extends Thread
{
public void run()
{
for(int j=1; j<=10; j=j+2)
system.out.println("Odd number: "+j);
}
}
public class Example7
{
public static void main(String args[ ])
{
Even el = new Even();
Odd ol = new Odd();
el.start();
ol.start();
}
}
```

Date: 24.05.2021

Signature:
Smilin