SLIATE

SRI LANKA INSTITUTE OF ADVANCED TECHNOLOGICAL EDUCATION

(Established in the Ministry of Higher Education, vide in Act No. 29 of 1995)

Higher National Diploma in Information Technology

First year, second Semester Examination – 2017

HNDIT1209 Object Oriented Programming

Instructions for Candidates:
Answer only five questions

No. of questions : 6
No. of pages : 3
Time: 3 hours

Question No	Expected Answer	Marks
1(I)	Collection of software components that enables a computer to run a Java program **(2 marks)**	Total 2
1(II)	Compiler (javac),Interpreter/loader (java),Java Runtime Environment (JRE) Archiver (jar),Documentation generator (javadoc),Any Other tools needed in Java development.	Total 4
1(III)	Explicit casting is required when assigning a larger data type to smaller data type.	Total 3
1(iv)(a)	S L I T E HNDIT	Total 2
1(iv)(b)	20 21	Total 2
1(iv)(c)	3 1	Total 2
1(v)(a)	True	Total 2
1(v)(b)	False	Total 2
1(v)(c)	True	Total 2
1(v)(d)	False	Total 2
1(v)(e)	True	Total 2

2(I)	A programme construct which execute same set of instructions repetitively	Total 2
2(II)	a. In while loop, if the condition is true then only statements in that loop will be executed. In do while loop statements within the do block are always executed at least once. b. In while loop, condition is tested at the beginning of the loop. In do while loop, condition is tested at the end of the loop **Any of the above (2 marks)**	Total 4
2(III)	Reduce code, save programmer time, able to use same variable repetitively, easy to design the structure of logic	Total 3
2(IV)	a. 01234 b. 01234	Total 2

2(V)	class SwitchTest
	{
	public static void main(String args[])
	{
	char operator='-'; // Any operator or default
	switch(operator)
	{
	case '+':
	System.out.println("Addition");break;
	case '-':
	System.out.println("Substraction");break;
	case '*':
	System.out.println("Multiplication");break;
	case '/':
	System.out.println("Devision");break;
	default:
	System.out.println("Not Allowed");
	}
	}
	}
		Total 5

3(I)	An *array* is a container object that holds a fixed number of values of a single type	Total 2

3(II)	values=new double[6]
	values[0]=24.0;
	values[1]=35.5;
	values[2]=67.4;
	values[3]=65.0;
	values[4]=33.0;
	values[5]=86.0;
	
	Or
	
	double values[]={24.0,35.5,67.4,65.0,33.0,86.0};
		Total 4

3(III)	class A{
	public static void main (String args[]){
	double values[]={1.0,2.0,3.0,4.0,5.0,6.0};
	int i=1; // **(1 mark)**
	while(i<values.length){ // or while (i<6) **(2 marks)**
	System.out.println(values[i]); // **(1 mark)**
	i=i+2; // **(1 mark)**
	}
	}
	}
		Total 3

3(IV)	enum season{WINTER,SUMMER,SPRING,FALL}	Total 6
3(V)	0000000000 111111111 22222222 3333333 444444 55555 6666 777 88 9	Total 5

4(I)	**Category** or **group** of things that have the same attributes and the same behaviors	Total 2

4(II)	class Employee
	{
	protected String empNo,empName;
	protected double basicSalary;
	}
		Total 4

4(III)	Employee(String empNo, String empName, double basicSalary)
	{
	This. empNo= empNo;
	This. empName = empName;
	This. basicSalary = basicSalary;
	}
		Total 3

4(IV)	Class_Laborer extends Employee
	{
	int OT_Hours;
	calNetSalary()
	{
	NetSal= basicSalary - (basicSalary *10%) + (OT_Hours *750)
	}
	}
		Total 6

4(V)	Title : Information Technology Subject :Java Hours :3	Total 5

5(I)	**polymorphism** is a feature that allows you to provide a single interface to varying entities of the same type.	Total 2
5(II)	Overloading Performed within class Parameter must be different Return type can be same or different Overriding Occurs in two classes – IS-A relationship Parameter must be same Return type must be same or covariant	Total 4
5(III)	It is a collection of abstract methods, which must be implemented by a class for inheriting interface functionality	Total 3
5(IV)	• Interface Account encapsulate double type field amount. • Value of Amount can't be changed since it is final by default • Amount and its value 100000.00 is inherited to FixedAccount and SavingAccount • Interface Account encapsulate abstract method interestRate • class FixedAccount implements interface Account to implement abstract method interestRate. • class SavingAccount implements interface Account to implement abstract method interestRate	Total 6

5(V))	FileOutputStream fout=new FileOutputStream("Data.txt");
	String s="Name ";
	byte b[]=s.getBytes();
	fout.write(b);
	fout.close();
		Total 5

6(I)	API for providing a graphical user interface (GUI) for Java programs. It provides lightweight components such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser	Total 2
6(II)	setBounds Specify the position and size of the a GUI component **(1 mark)** setBounds(int x, int y, int width, int height) Then (x, y) is the coordinate of the upper-left corner of the component. x is the number of pixels from the left of the screen and y is the number of the pixels from the top of the screen.width and height used to specify the size of the component **(1 mark)** **setSize** is used to set the size of the component **setSize**(int width, int height)	Total 4
6(III)	Procedure or function in a computer program that waits for an **event** to occur, and	Total 3

6(IV)	import javax.swing.*;
	class GUI1 {
	public static void main(String[] args) {
	JFrame f=new JFrame();
	
	JLabel l1,l2,l3;
	l1=new JLabel("Marks 1");
	l1.setBounds(50,0,100,30);
	l2=new JLabel("Marks 2");
	l2.setBounds(50,25,100,30);
	l3=new JLabel("Grade");
	l3.setBounds(50,50,100,30);
	
	JTextField t1,t2,t3;
	t1=new JTextField();
	t1.setBounds(200,5,100,20);
	t2=new JTextField();
	t2.setBounds(200,30,100,20);
	t3=new JTextField();
	t3.setBounds(200,55,100,20);
	
	JButton b=new JButton("Display");
	b.setBounds(130,100,75, 20);
	
	f.add(l1); f.add(l2); f.add(l3);
	f.add(t1); f.add(t2); f.add(t3);
	f.add(b);
	
	f.setSize(400,200);
	f.setLayout(null);
	f.setVisible(true);
	}
	}
		Total 6

	Or
	import javax.swing.*;
	class Example2 extends JFrame {
	JFrame f;
	Example2(){
	f=new JFrame();
	JLabel l1,l2,l3;
	l1=new JLabel("Marks 1");
	l1.setBounds(50,0,100,30);
	l2=new JLabel("Marks 2");
	l2.setBounds(50,25,100,30);
	l3=new JLabel("Grade");
	l3.setBounds(50,50,100,30);
	
	JTextField t1,t2,t3;
	t1=new JTextField();
	t1.setBounds(200,5,100,20); //
	t2=new JTextField();
	t2.setBounds(200,30,100,20);
	t3=new JTextField();
	t3.setBounds(200,55,100,20);
	
	JButton b=new JButton("Display");
	b.setBounds(130,100,75, 20); //
	
	f.add(l1); f.add(l2); f.add(l3);
	f.add(t1); f.add(t2); f.add(t3);
	f.add(b);
	
	f.setSize(400,200);
	f.setLayout(null);
	f.setVisible(true); }
	
	public static void main(String[] args) {
	new Example2();
	}
	}	

6(V)	g.setColor(Color.blue) -Set the graphic color to blue **(1 mark)** g.drawString("SLIATE",200, 50) -Draw a blue color String "SLIATE" **(1 mark)** g.drawRect(70,100,50,50) -Draw a blue color square/rectangle, width of 50px and height of 50px **(2 mark)** g.setColor(Color.red)- Set the current graphic color to red **(1 mark)** g.drawLine(10,30,10,300)-Draw a red color line between the point (10, 30) and (10,300) **(1 mark)**	Total 5