Internal Assessment Test 1 – July 2022

Scheme & Model Solution

<table>
<tr><td colspan="2">Sub: Object Oriented Concepts</td><td>Sub Code: 18CS45</td><td>Sem/Branch: IV / CSE</td><td>Sections: A,B,C</td></tr>
<tr><td></td><td></td><td></td><td>MARKS</td><td>CO</td><td>RBT</td></tr>
<tr><td>Question</td><td>1a</td><td>Differentiate between procedure-oriented programming and object-oriented programming.</td><td>5</td><td>CO1</td><td>L2</td></tr>
<tr><td>Scheme</td><td></td><td>Difference – 6 points (3M)
1 Example for each (2M)</td><td>3+2</td><td></td><td></td></tr>
<tr><td>Solution</td><td></td><td>
<table>
<tr><th>#</th><th>Procedure Oriented Programming</th><th>Object Oriented Programming</th></tr>
<tr><td>1</td><td>Program is divided into small parts called functions</td><td>Program is divided into parts called objects</td></tr>
<tr><td>2</td><td>Focus is on procedures. The code is centered around procedures.</td><td>Focus is on data. Code is centered around data.</td></tr>
<tr><td>3</td><td>Procedures are dissociated from data and are not part of it.</td><td>Procedures are bound to the data.</td></tr>
<tr><td>4</td><td>Data is not secure. Compilers that implement the procedure-oriented programming system do not prevent unauthorized functions from accessing/manipulating structure variables.</td><td>Enables data security by throwing compile time errors against piece of code that violate the prohibition.</td></tr>
<tr><td>5</td><td>Data is not initialized</td><td>Provides guaranteed initialization of of data. Programmers can ensure a guaranteed initialization of data members of structure variables to the desired values.</td></tr>
<tr><td>6.</td><td>Overloading is not supported</td><td>Supports overloading of operators and functions</td></tr>
</table>

example program in C :

#include <stdio.h>
int main() {

int number1, number2, sum;

printf("Enter two integers: ");
scanf("%d %d", &number1, &number2);

// calculating sum
sum = number1 + number2;

printf("%d + %d = %d", number1, number2, sum);
return 0;
}
</td><td></td><td></td><td></td></tr>
</table>

		example program in C++.
		#include <iostream>
		using namespace std;
		
		int main() {
		
		int first_number, second_number, sum;
		
		cout << "Enter two integers: ";
		cin >> first_number >> second_number;
		
		// sum of two numbers in stored in variable sumOfTwoNumbers
		sum = first_number + second_number;
		
		// prints sum
		cout << first_number << " + " << second_number << " = " << sum;
		
		return 0;
		}			

Question	1b	Explain how one can bridge two classes using friend Function. Write a C++ program to find largest among two numbers using friend Function. Assume two variables are present in two different class.	5	CO2	L3
Scheme		Explanation (1M) Program (4M)	1+4		

Solution		Friend function can be used as bridge between two classes. To bridge two classes with a function, the function should be declared as a friend to both the classes. Then the friend function can access private data of both the classes.
		
		#include <iostream>
		using namespace std;
		
		class B ; // Forward declaration
		
		//void largest (A, B);
		
		class A
		{
		private:
		int a;
		
		public:
		A()
		{
		a = 100;
		}			

```
                friend void largest(A,B);
};


class B
{
        private:
                int b;
        public:
                B()
                {
                        b = 200;
                }
                friend void largest(A,B);
};

void largest (A Aobj, B Bobj)
{

if(Aobj.a > Bobj.b)
        cout << "largest of private members
of class A and B = " << Aobj.a );

else
        cout << "largest of private members
of class A and B = " << Bobj.b);
        }

int main()
{
        A A1;
        B B1;
        largest (A1, B1);
        return 0;
}
```

Question	2a	Explain the use of scope resolution operator with an example program.	5	CO1	L2
Scheme		Explanation/Justification (2M) Program (3M)	2+3		

Solution		The scope resolution operator (::) is used for several reasons. For example: If the global variable name is same as local variable name, the scope resolution operator will be used to call the global variable. It is also used to define a function outside the class.
		
		`#include<iostream>`
		`using namespace std;`
		
		`class A`
		`{`
		`public:`			

		// Only declaration
		void fun();
		};
		
		// Definition outside class using ::
		void A::fun()
		{
		cout << "fun() called";
		}
		
		int main()
		{
		A a;
		a.fun();
		return 0;
		}			

Question	2b	Write a C++ program to overload the member function area() to find area of rectangle and area of triangle	5	CO2	L3
Scheme		Program (5M)	5		

Solution		#include<iostream>
		using namespace std;
		int area(int,int);
		float area(float,float);
		int main()
		{
		int l,b;
		float bs,ht;
		cout<<"Enter length and breadth of rectangle:";
		cin>>l>>b;
		cout<<"Enter base and height of triangle:";
		cin>>bs>>ht;
		cout<<"\nArea of rectangle is "<<area(l,b);
		cout<<"\nArea of triangle is "<<area(bs,ht);
		return 0;
		}
		int area(int l,int b)
		{
		return(l*b);
		}
		float area(float bs,float ht)
		{
		return((bs*ht)/2);
		}			

Question	3a	Write a C++ program using static data member to count the number of objects created.	5	CO2	L3
Scheme		Program (5M)	5		
Solution		# include<iostream> using namespace std; class A { int code; static int count;			

```
public:
A ()
{
count++;
}
void showcount(void)
{
cout << "The number of objects created is " << count << endl;
}
};
int A :: count;
int main()
{
A obj1, obj2, obj3,obj4;
obj1.showcount();
}
```

Question	3b	Differentiate between parameterized constructor and copy constructor.	5	CO1	L2
Scheme		Difference- 4 points (2M) 1 Example program for each (3M)	2+3		
Solution					

#	Parameterized Constructor	Copy Constructor
1	A Parameterized constructor is a member function that is used to initialize the various data elements of different objects with different values when they are created.	A copy constructor is a member function which initializes an object using another object of the same class.
2	It is useful when we want to provide different values to the objects.	It is helpful when we want to copy a complex object that has several fields, or when we want to make a deep copy of an existing object.
3	It carries a regular variable parameter in its prototype.	It carries an object parameter in its prototype. In copy constructor we need to use reference of object.

4	**Syntax:**
	
	name_of_class (variables)
	{ //Code for constructor
	}
		Syntax:
		
		Class_name(const class_name &object_name)
		{
		// body of the constructor
		}

<u>Parametrized constructor Example Program:</u>

```
#include<iostream>
using namespace std;
class A {
   private:
   int a, b;
   public:
   A(int a1, int b1)
   {
      a = a1;
      b = b1;
   }
   int getA()
   {
      return a;
   }
   int getB()
   {
      return b;
```

```
    } };
int main()
{
  A obj1(10, 15);

 cout << "a = " << obj1.getA() ;
cout << ", b = " << obj1.getB();

  return 0;
}
```

copy constructor Example Program:

```
int main()
{
  A obj1(10, 15);

 cout << "a = " << obj1.getA() ;
cout << ", b = " << obj1.getB();

  return 0;
}
void Display()
{
cout<<"\nValues :"<< copy_a <<"\t"<< copy_b;
}
};
int main()
{
copycon obj(10,20);
copycon obj2=obj; //Copy Constructor
cout<<"\nI am parameterized Constructor";
obj.Display();
// Constructor invoked.
cout<<"\nI am copy Constructor";
obj2.Display();
return 0;
}
```

Question	4a	List the rules for mentioning default arguments in a function declaration. With an example program show its usage.	5	CO1	L2
Scheme		List of Rules-4points (2M) Example Program showing usage (3M)	2+3		
Solution		Rules: • Only the last argument must be given default value. You cannot have a default argument followed by non-default argument. • If you default an argument, then you will have to default all the subsequent arguments after that. • You can give any value a default value to argument, compatible with its datatype.			

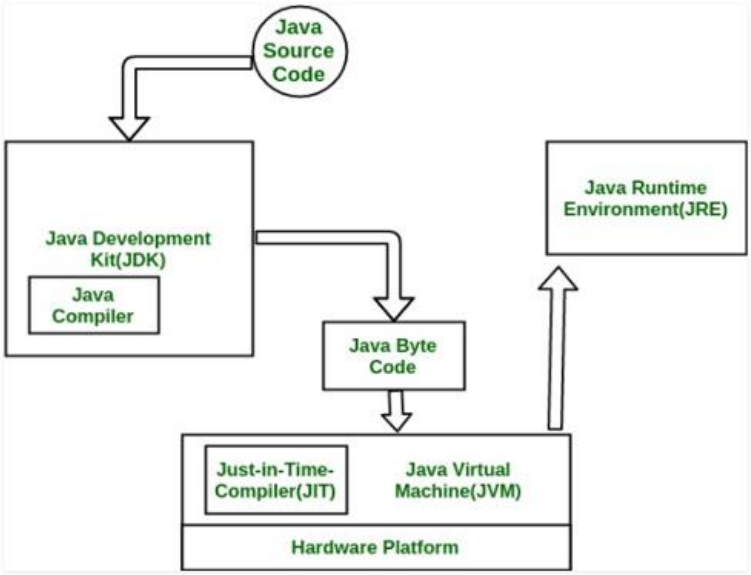
		• Example:
		
		int mul (int i, int j=5, int k=10); //legal.
		int mul (int i=5, int j); //illegal.
		int mul (int i=0,int j, int k=10); //illegal.
		int mul (int i=2, int j=5, int k=10); //legal.
		
		#include <iostream>
		using namespace std;
		
		// A function with default arguments,
		// it can be called with
		// 2 arguments or 3 arguments or 4 arguments.
		int sum(int x, int y, int z = 0, int w = 0)
		//assigning default values to z,w as 0
		{
		return (x + y + z + w);
		}
		// Driver Code
		int main()
		{
		// Statement 1
		cout << sum(10, 15) << endl;
		
		// Statement 2
		cout << sum(10, 15, 25) << endl;
		
		// Statement 3
		cout << sum(10, 15, 25, 30) << endl;
		return 0;
		}			

Question	4b	Write a C++ program to create a class called Employee with data members name , age and salary. Display at least 10 employee information	5	CO2	L3
Scheme		Program (5M)	5		

Solution		#include<iostream>
		using namespace std;
		class Employee
		{
		private:
		int age;
		char name[20];
		int salary;
		
		public:
		Employee()
		{
		salary = 1000;
		}
		void GetData();
		void DispData();
		};			

		void Employee::GetData()
		{
		cout<<"Enter the employee age: ";
		cin>>age;
		cout<<"Enter the employee name: ";
		cin>>name;
		cout<<"Enter the employee salary: ";
		cin>>salary;
		}
		void Employee::DispData()
		{
		cout<<endl<<age<<"\t"<<name<<"\t"<<salary<<"\t;
		}
		int main()
		{
		Employee e[10];
		cout<<"Enter the employee information:"<<endl;
		for(int i=0;i<10;i++)
		{
		e[i].GetData();
		}
		cout<<endl<<"The employee information is:";
		cout<<endl<<"EmpID \t Name \t Bsalary \t Allowance";
		for(int i=0; i<10;i++)
		{
		e[i].DispData();
		}
		}			

Question	5a	Briefly discuss about JVM, JRE, JDK	5	CO1	L2
Scheme		Explanation of each (1.5 M) Diagram (0.5M)	4.5+0.5		
Solution		<u>JAVA VIRTUAL MACHINE</u> JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode. JVMs are available for many hardware and software platforms. JVM, JRE, and JDK are platform dependent because the configuration of each OS is different from each other. However, Java is platform independent. There are three notions of the JVM: specification, implementation, and instance. The JVM performs the following main tasks: Loads code Verifies code Executes code Provides runtime environment			

		JAVA RUNTIME ENVIRONMENT JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime. JAVA DEVELOPMENT KIT JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools. The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.  			
Question	5b	Write a Java program to create a class circle which contains data item radius , read(), display(),area() to read user input,to display the input and find area of circle.	5	CO2	L3
Scheme		Program (5M)	5		

Solution		public class circle
		{
		public static void main(string[] args)
		{
		float r, area;
		Scanner sc=new Scanner(System.in);
		void read()
		{
		System.out.println(“enter radius value”);
		r=sc.nextFloat();
		}			

		void area()
		{
		area=3.14*r*r;
		}
		void display()
		{
		System.out.println(“Area=”+area);
		}
		
		}
		}			

Question	6a	Write a Java program to print sum of n numbers in a 1-D array.	5	CO1	L2
Scheme		Program (5M)	5		

Solution		class Test {
		static int arr[] = { 12, 3, 4, 15 };
		
		// method for sum of elements in an array
		static int sum()
		{
		int sum = 0; // initialize sum
		int i;
		
		// Iterate through all elements and add them to sum
		for (i = 0; i < arr.length; i++)
		sum += arr[i];
		
		return sum;
		}
		
		// Driver method
		public static void main(String[] args)
		{
		System.out.println("Sum of given array is "
		+ sum());
		}
		}			

Question	6b	Explain the operations of the following operators with example. i) short circuit operators ii) >>>	5	CO1	L2
Scheme		2 operator explanation with program (2*2.5)	2.5+2.5		
Solution		i) In Java logical operators, if the evaluation of a logical expression exits in between before complete evaluation, then it is known as Short-circuit. A short circuit happens because the result is clear even before the complete evaluation of the expression, and the result is returned AND(&&) short circuit: In the case of AND, the expression is evaluated until we get one false result because the result will always be false, independent of the further conditions. If there is an expression			

with &&(logical AND), and the first operand itself is false, then a short circuit occurs, the further expression is not evaluated, and false is returned.

```
import java.io.*;

class ShortCirAND {
   public static void main(String arg[])
   {

      // Since first operand is false
      // and operator is &&,
      // Evaluation stops and
      // false is returned.
      if (false && true && true)
{
         System.out.println("This output "
                        + "will not "
                        + "be printed");
      }
      else
{

         System.out.println("This output "
                        + "got printed actually, "
                        + " due to short circuit");
      }

      // Whole expression will be evaluated,
      // as no false is encountered
      // before last condition
      // Therefore no Short circuit

 if (true && true && true)
{
         System.out.println("This output "
                        + "gets print"
                        + " as there will be"
                        + " no Short circuit");
      }
      else
{

         System.out.println("This output "
                        + "will not "
                        + "be printed");
      }
   }
}
```

ii) In Java, the operator '>>>' denotes unsigned right shift operator and always fill 0 irrespective of the sign of the number.

```
class GFG {

    // main driver method
    public static void main(String args[])
    {

        // x is stored using 32 bit 2's complement form.
        // Binary representation of -1 is all 1s (111..1)
        int x = -1;

        // The value of 'x>>>29' is 00...0111
        System.out.println(x >>> 29);

        // The value of 'x>>>30' is 00...0011
        System.out.println(x >>> 30);

        // The value of 'x>>>31' is 00...0001
        System.out.println(x >>> 31);
    }
}
```