**FACULTY OF ENGINEERING & TECHNOLOGY,**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Cycle Test – I**

**Academic Year: Aug-Dec 2020**

**Program offered: B.Tech (ECE) Year / Semester: II/III**

**Max. Marks: 25 Duration: 1 hr**

**Date of Exam: 25 /08/2020 Time of Exam: 2.30 pm – 3.30 pm**

**Course Code and Title: 18CSC202J – Object Oriented Design and Programming**

**25 Mcqs (4 difficult 8 moderate 13 easy)**

**1. The programming language that uses stdio.h as the header file has the feature of**

**i) Global data and local data**

**ii) Modules and sub modules**

**iii) Data reusable**

**iv) Data hiding**

a) i and iii only

b) ii,iii and iv only

c) i only

d) i and ii only

**2. The input and output operations involve the transfer of bytes known to be**

a) Streams

b) Pointers

c) Objects

d) Class

**3. The value of a data type when converted into the another data type is known to be**

a) Polymorphism

b) Class

c) Casting

d) Data reusable

**4. The member function used to assign the value to the variables**

a) Constructor

b) Destructor

c) inline function

d) friend function

**5. The function which supersedes the data hiding feature in Object Oriented Programming is**

a) Static function

b) Friend function

c) Constant variable

d) Constructor

**6. The class is defined to be**

**i) Blue print**

**ii) Declares features and behavior**

**iii) Defines the members**

**iv) Allocates space for the variables**

a) i and iv

b) i ,ii,iii and iv

c) ii and iv

d) I,ii and iii

**7. The design perspective of Object Oriented concepts for the structural and behavioral aspect lays the diagrams of**

a) Use case diagram and object diagram

b) Use case diagram and class diagram

c) Class diagram and object diagram

d) Class diagram, object diagram and usecase diagram

**8. The notation in UML diagram that combines related classes of objects into a distinct, more general class**

a) Association

b) Composition

c) Generalization

d) Multiplicity

**9. A printer is made up of many components like scanner, Xerox with black or color, paper settings etc… if the processor represents all the functioning of printer, xerox shows the duplicate of the copy and the printer is represented as a whole. Which among the following have highest level of abstraction?**

a) Xerox

b) Scanner

c) Motherboard

d) Printer

**10. State the inline functions in the given code:**

**#include<iostream.h>**

**class student**

**{**

**int regno;**

**char name[10];**

**public:**

**void getdata()**

**{**

**cin>>regno;**

**cin>>name;**

**}**

**void processdata();**

**void displaydata();**

**};**

**inline void student::processdata()**

**{**

**int m1,m2,m3;**

**cin>> m1>>m2>>m3;**

**cout <<”Total =”<<m1+m2+m3;**

**}**

**void student::displaydata()**

**{**

**cout<<”Regno:”<<regno;**

**cout<<”Name:”<<name;**

**}**

**void main()**

**{**

**student s;**

**s.getdata();**

**s.displaydata();**

**s.processdata();**

**}**

a) displaydata()

b) displaydata() and processdata()

c) getdata() and displaydata()

d) getdata() and processdata()

**11. If we don’t use classes in a program**a) Features of function overloading, static members and inline functions can be used

b) Features of inheritance and encapsulation can be implemented

c) Features of function overloading and encapsulation can be implemented

d) Features of inheritance and inline functions can be used.

**12. The notation of UML that is represented at the end of an association, indicates the number of instances of one class linked to number of instances of other class**

a) Multiplicity

b) Role

c) Name

d) Dependency

**13. The object diagram in UML that emphasis on**

a) The member functions of the class

b) The instance of the class

c) The access modifiers of the class

d) The access specifiers and functions of the class

**14.**

**The following program fragment     
void main ( )  
{  
 int v=10;  
 int &a=v;  
 cout << &a <<  &v;  
}**

a) Prints 10 and address of v

b) Prints address of a and 10

c) Prints address of a or v twice

d) Compilation error

**15. State the accessibility of the following code:**

**class sample**

**{**

**int a,b;**

**public:**

**int c,d;**

**void get();**

**void display();**

**}**

a) a and b are public

b) a and b are private

c) a and b are protected

d) a and b are public and protected

**16. The notations that can be used in Use Case diagram are**

**i) describing the portion of action/function which is common across more than one use case, ii) adding more functions to the existing use case, iii) scope of the system iv) actions of the system**

a) include,extend,usecase,system boundary

b) extend,include,system boundary,usecase

c) include,extend, system boundary,usecase

d) extend, include, usecse, system boundary

**20. The result obtained in the fragment of the following program**

**int a=10;**

**static int b=5;**

**int main()**

**{**

**{**

**b=b+a;**

**int c=10;**

**a=b+a;**

**}**

**cout<<b<<”\t”;**

**cout<<c;**

**return 0;**

**}**

a) 15 10

b) 15 0

c) Compiler error

d) 5 10

**21. To produce the result as true insert the code to the fragment of the following program**

**int a = 5;**

**int b = 5;**

**string c ="hello";**

**string d ="Hello";**

**if( // insert the code )**

**cout<<"true";**

**else**

**cout<<"false";**

a) a=b and c!d

b) a=b or c!d

c) a==b and c!=d

d) a==b and c==d

**22. State the appropriate class code for the following program**

**class ABC**

**{**

**int a,b,c;**

**public:**

**ABC();**

**ABC(int ,int ,int );**

**};**

**// insert code here**

a) void main() { ABC obj; ABC obj(2,3,4) }

b) int main() { ABC obj; ABC obj(2,3,4); return 0;}

c) ABC::ABC(){ } ABC::ABC(int x,int y,int z){ a=x; b=y; c=z; }

d) void ABC::ABC(){ } void ABC::ABC(int x,int y,int z){ a=x; b=y; c=z; }

23. **State the number of errors in the following program**

**class example**

**{**

**int a,b,c;**

**public:**

**int d;**

**void get()**

**{**

**cin>>a>>b>>c>>d;**

**}**

**void display();**

**}**

**example::display()**

**{**

**cout<<example::a<<b<<c<<d;**

**}**

**int main()**

**{**

**example obj;**

**obj.get();**

**cout<<”global data”<<obj.d;**

**cout<<example::d;**

**cout<<d;**

**cout<<obj.display();**

**}**

a) 3

b) 4

c) 5

d) 6

**24. The output for the following code:**

**class ABC**

**{**

**int i;**

**public:**

**ABC()**

**{**

**i=0;**

**cout<<"c\tc"<<endl;**

**}**

**~ABC()**

**{**

**cout<<".."<<endl;**

**}**

**};**

**void f()**

**{**

**static ABC obj,obj2,ob3;**

**}**

**int main()**

**{**

**int x=0;**

**if(x==0)**

**{**

**f();**

**}**

**cout<<"#"<<endl;**

**return(0);**

**}**

a)

c\tc

#

..

b)

c c

#

..

c)

c c

c c

c c

#

..

..

..

d)

c c

..

#

**25) state the code for defining and printing the value of pointer**

**i) int x, \*p;**

**ii) &p=x;**

**iii) p=&x;**

**iv) cout<<p;**

**v) cout<<\*p;**

a) i,ii,iv

b) i,iii,v

c) ii,iv

d) ii,v