Total No. of Questions – [ ] Total No. of Printed Pages 07

G.R. No.	

Paper Code - U218 - 126 (T2)

OCTOBER 2018/ IN-SEM (T1)
S. Y. B. TECH. (COMPUTER) (SEMESTER - I)
COURSE NAME: OBJECT ORIENTED PROGRAMMING
COURSE CODE: (CSUA21176)

(PATTERN 2017)

Time: [**1 Hour**] [Max. Marks: **30**]

(*) Instructions to candidates:

1) **Answer Q.1 OR Q.2 and Q.3 OR Q.4.**
2) **Figures to the right indicate full marks.**
3) **Use of scientific calculator is allowed**
4) **Use suitable data where ever required**

Q.1) a) int main()

2 marks proper use of pointer noatation
4 marks Proper logic

```
{
float flarr[100];
char ch;
int num = 0;
Do
{
cout << "Enter number: ";
cin >> *(flarr+num++);
cout << "Enter another (y/n)? ";
cin >> ch;
}
while(ch != 'n');
float total = 0.0;
for(int k=0; k<num; k++)
total += *(flarr+k);
float average = total / num;
cout << "Average is " << average;
return 0;
}
```

1b) class and object 2 marks
Proper logic of program 4 marks

```
#include<iostream>
#include<cstring>
using namespace std;
class media
 {
protected:
        char title [50];
        float price;
public:
        media(char *s,float a)
        {
        strcpy(title,s);
        price=a;
        }
        virtual void display()
        {
        }
 };

class book:public media
 {
        int pages;
public:
        book(char *s,float a,int p):media(s,a)
        {
        pages=p;
        }
        void display();
 };

class tape:public media
 {
        float time;
public:
        tape(char *s ,float a,float t):media(s,a)
        {
        time=t;
        }
        void display();

 };
```

```
void book::display()
 {
 cout<<"Title"<<title;
 cout<<"Pages"<<pages;
 cout<<"Price"<<price;
 }
void tape::display()
 {
 cout<<"Title"<<title;
 cout<<"Play Time"<<time<<"mins";
 cout<<"Price"<<price;
 }

int main()
{
char *title =new char[30];
float price,time;
int pages;
cout<<"Enter book details";
cout<<"Title";
cin>>title;
cout<<"price";
cin>>price;
cout<<"pages";
cin>>pages;
book book1(title,price,pages);
cout<<"Enter book details";
cout<<"Title";
cin>>title;
cout<<"price";
cin>>price;
cout<<"play time";
cin>>time;
tape tape1(title,price,pages);
media *list[2];
list[0]=&book1;
list[1]=&tape1;
cout<<"Media details";
cout<<"BOOK";
list[0]->display();
```

```
cout<<"Tape";
list[1]->display();
}
```

C)Definition of virtual Base class 2 Marks
Example of Multiple inheritance 2 Marks

OR

Q.2) a) Definition of Abstract class 2 Marks
Along with Example of single inheritance and Pure virtual Function 4Marks

b) new and delete with syntax of each 2 marks
Example of each 2 marks each

c) Definition 1 mark
Example 3 marks

Q.3) a) class 2 mark
4 Exception 1 mark each

```
#include<iostream>
using namespace std;
class exc1
{
 public:
 int age,income;
 string city;
 int a;
 void display()
 {
  try
   {
    cout<<"Enter the age: \n";
    cin>>age;
    if(age>=18 && age<=55)
     {
      cout<<"Age is : "<<age<<"\n";
     }
    else
     throw(age);
```

```
 }

 catch(int age)
 {
  cout<<"Caught Exception\n";

 }
}

void display1()
{
 try
 {
   cout<<"Enter the income: \n";
   cin>>income;
   if(income>=50000 && income<=100000)
    {
      cout<<"Income is : "<<income<<"\n";
    }
   else
   throw(income);
 }
 catch(int income)
 {
  cout<<"Caught Exception\n";
 }
}

void display2()
{
 try
  {
   cout<<"Enter the city: \n";
   cin>>city;
  if(city=="Pune" || city=="Mumbai" || city=="Banglore" || city=="Chennai")
   {
    cout<<"City is : "<<city<<"\n";
   }
  else
   throw(city);
 }
```

```
  catch(string city)
   {
    cout<<"Caught Exception\n";
   }
 }

 void display3()
 {
  try
   {
    cout<<"Enter type of vehicle 2 or 4 : \n";
    cin>>a;
    if(a==4)
     {
      cout<<"Vehicle is :"<<a<<" wheeler \n";
     }
   else
   throw(a);
   }
   catch(int a)
    {
     cout<<"Caught Exception\n";
    }
  }

};
int main()
{
exc1 obj;
obj.display();
obj.display1();
obj.display2();
obj.display3();
return 0;
}
```

Q.3) b)

```
#include <iostream>
        using namespace std;
        template<class x1,class x2>void swapargs(x1 &a,x2 &b)
```

3 Marks

```
{
cout << "Inside swapargs Generic specialization.\n";
x1 temp;
temp = a;
a = b;
b = temp;
}

int main()
```

3 Marks

```
{
swapargs(i, j; //INT
cout << "Swapped i, j: " << i << ' ' << j << '\n';
swapargs(x, y); // FLOAT
cout << "Swapped x, y: " << x << ' ' << y << '\n';
swapargs(a, b); // CHARACTER
cout << "Swapped a, b: " << a << ' ' << b << '\n';
return 0;
}
```

c) 2 mark each

Q.4) a) Try catch and throw block 3 marks

Program with Proper Logic 3 marks

b)2 marks each [4 marks]

c) 1 mark each [4 marks]