

B. INFO. TECH 2<sup>ND</sup> YEAR 1<sup>ST</sup> SEMESTER EXAMINATION, 2018  
(1<sup>ST</sup> Semester)

OBJECT ORIENTED PROGRAMMING

Time: Three hours

Full Marks: 100

*Answer Question No. 1 and any THREE from the rest*

1. Answer the following questions.

(a) Consider the following template definition for a list.

There are many structures that have two data members.

```
struct Point {
    float x;
    float y;
};

struct Money {
    int Rupee;
    int paisa;
}
```

Define a template having the following functionalities:

- Constructor having one or two arguments of given types, and a copy-constructor.
- Methods to compare two objects of same type.
- Method to swap between two objects.

Do not use `std::pair`.

2+3+2+3

(b) Consider an exception class that holds a string. The string contains the name of the function from which the exception object is thrown (or re-thrown), and the string compatible values of the actual parameters. The function that catches it again throws the exception object after appending name of the function which catches it (and from which the exception object is re-thrown), and the string compatible values of the actual parameters. This process continues till the main function catches it where the exception object is printed as a whole.

Suppose that the following is the call stack when an exception object is thrown.

`main() -> f(int, char) -> g(int) -> h(double)`.

The first exception object was thrown at `h()`.

Define the exception class. Write the minimum code in functions `main()`, `f()`, `g()`, and `h()` to explain the working of the idea. You may use `String Stream` class for writing onto the string. OR You may use a `String` class with appropriate constructors and "append" method. Do not define `String` class.

2+ 3+5

(c) Suppose that a text file is to be read and a table is to be constructed such that each row in that table contains a letter in the English Alphabet (a-z, A-Z) and the number of times the letter occurs in the text file. This table is then written onto a file. Write a C++ program for this purpose. Define the classes for the file reader

and the table along with their member functions. Also, define a *main* function. Use File Streams.

7+6+7

2. Mention two operators, which cannot be overloaded. Mention two operators, which cannot be overloaded as friend functions and why?

1+3

Implement a class "String" in C++. The class "String" has two data members for storing the array of characters and length. Write appropriate methods to overload the following operators: (i) "+" operator (for concatenation), (ii) "=" operator, (iii) "==" operator for equality comparison, (iv) "<<" operator, (v) ">>" operator, and (vi) the "[" operator.

Do not use any built-in class.

4+4+2x4

3. Answer the following questions.

- (i) Why is a virtual destructor required?
- (ii) Give an example where private inheritance is used.
- (iii) Why do we use friend functions or friend classes?
- (iv) What is the problem of multiple occurrence of the base class in the context of multiple inheritance? How is this problem solved?
- (v) State three advantages of new over malloc.
- (vi) In what ways 'throw' is different than a 'new' (of an exception object) followed by 'return'?
- (vii) What is the syntax of a copy constructor? When is a copy constructor invoked?

2+2+1+(3+3)+3+3+(1+2)

4. Answer the following questions.

- a) Suppose that a *Cell* contains a number of *Patterns*. A *Cell* object is a rectangle which is defined by its height, width and its position in terms of x-coordinate and y-coordinate of its left bottom point. Each *Pattern* object contains a code (whether it is a triangle or a rectangle or a trapezoid) and the position of its centre in terms of x-coordinate and y-coordinate with respect to the left bottom point of the containing *Cell* object. A *Cell* object may be a *SimpleCell* or an *ArrayCell*. An *ArrayCell* object has a count, and displacements along x and y directions. Suppose that an *ArrayCell* object has count 3, x-displacement 2, y-displacement 1 and the x-coordinate and y-coordinate of its left bottom point is (4,5). It means that there are 3 *SimpleCell* objects with the same set of patterns where the x-coordinate and y-coordinate of the left bottom point of the i-th *SimpleCell* object is (4+i\*2, 5+i\*1) for 0<=i<=2. Finally a *Block* has a number of *Cells*.

The classes *Block* and *Cell* has a method to count the total number of patterns in it. Define the classes *Block*, *Cell*, *Pattern*, *SimpleCell*, *ArrayCell* in C++.

5x3=15

- b) Give an example of a function prototype that uses default arguments. What are the restrictions of using default arguments in a function prototype? Can default arguments be used in the context of virtual functions?

1+2+2

5. Answer the following questions.

- (i) What is the syntax of reference to an integer pointer? In the context of parameter passing, explain the difference between passing a pointer or a reference to a pointer with an example.
- (ii) Why the methods to overload << operator always returns a reference?
- (iii) What is a namespace in C++? How do you define it? How do you use it?
- (iv) What is wrong with the following code?

```
class Employee {
public:
    Employee(string name);
// ...
};
class Manager {
public:
    Manager(string name, string sname);
    ~Manager();
private:
    Employee* _secy;
};

Manager::Manager(string name, string sname)
: Employee(name), _secy(new Employee(sname))
{}

Manager::~~Manager() { delete _secy; }
```

- (v) Consider the class String.  
Which of the underlined statements are correct?

```
class String {
public:
    char & operator [ ] (int pos) { return data[pos]; }
    const char & operator [ ] (int pos) { return data[pos]; }
private:
    char * data;
    .....
}
String s1 = "Hello";
cout << s[0];
s[0] = 'x';
const String cs = "World";
cout << cs[0];
cs[0] = 'x';
```

(1+3)+2+(1+1+2)+5+5