MUZAFFARPUR INSTITUTE OF TECHNOLOGY, MUZAFFARPUR

OBJECT ORIENTED PROGRAMMING

(051X01)

RAJEEV KUMAR

ASSISTANT PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY

CONTENTS

Department of Information Technology

Vision

To achieve global standard in quality of education, research & development in Information Technology by adapting to the rapid technological advancement to empowering the IT-industry with the wings of knowledge and power of innovation though knowledge creation, acquisition and dissemination for the benefit of Society and Humanity.

Mission

- To produce well-rounded, up to date, scientifically tempered, design oriented engineer and scientists capable of lifelong learning.
- To produce technologically competent and ethically responsible graduates through balanced and dynamic curriculum.
- To develop highly analytical and qualified IT engineers by imparting training on cutting edge technology professional ethics to make the nation as a knowledge power.
- To generate high quality knowledge resource in area of Information Technology and in emerging area to make valuable contribution in IT-Sector for social and economic development of nation and to make organized for identification, monitoring and control of objective attributes of quality for continuous enhancement of academic progress, infrastructure and ambience.

Information Technology Program Educational Objectives

After 4 to 5 years of graduation a B. TECH (IT) graduate would be able to
Graduates of the program will be employed in the computing profession, and will be engaged in learning, understanding, and applying new ideas and technologies as the field evolves.

Graduates with an interest in, and aptitude for, advanced studies in computing will have completed, or be actively pursuing, graduate studies in computing.

Graduates will be informed and involved members of their communities, and responsible engineering and computing professionals who take appropriate account, in their professional work, of such issues as privacy, security, copyright etc. in ways that are consistent with the ACM/IEEE Code of Conduct.

Information Technology Student Outcomes

Students who complete the B. TECH degree in IT will be able to:

an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
an ability to design and conduct experiments, as well as to analyse and interpret data;
an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
an ability to function on multi-disciplinary teams;
an ability to identify, formulate, and solve engineering problems;
an understanding of professional, ethical, legal, security and social issues and responsibilities;
an ability to communicate effectively with a range of audiences;
an ability to analyse the local and global impact of computing on individuals, organizations, and society;
a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
a knowledge of contemporary issues;
an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
an ability to analyse a problem, and identify and define the computing requirements appropriate to its solution;
an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modelling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices;
an ability to apply design and development principles in the construction of software systems of varying complexity.

Course Description

- This course is part of the Scientific Computing series.
- This course provides an introduction to object oriented programming (OOP), aimed at those who have no prior experience of object oriented programming.
- The concepts of object oriented programming are demonstrated in Python.
- By the end of the course, attendees should have a basic familiarity with these concepts and be able to assess whether the object-oriented paradigm is appropriate for their programming tasks.
- As this course is part of the Scientific Computing series, most of the examples in the presentation are of greatest relevance to scientific programming.

Course Objectives

- Perform object oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O. and other standard language constructs.
- Demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.
- Demonstrate ability to implement one or more patterns involving realization of an abstract interface and utilization of polymorphism in the solution of problems which can take advantage of dynamic dispatching.
- Learn syntax, features of, and how to utilize the Standard Template Library.
- To study the concepts of Assembler, Macro Processor, Loader and Linker

Course Outcomes

- Explain the steps in creating an executable program for a computer, including the intermediate representations and their purpose.
- Manipulate binary patterns and understand the use of binary to represent numbers.
- Apply good programming style and understand the impact of style on developing and maintaining programs.
- Effectively use a version control system and the Linux command line tools for incremental development.
- Explain the benefits of object oriented design and understand when it is an appropriate methodology to use.
- Design object oriented solutions for small systems involving multiple objects.
- Implement, test and debug solutions in C++.
- Identify the relative merits of different algorithmic designs.
- Independently find and interpret discipline related documentation.
- Explain the relevance of ethics in the context of Software Engineering.

Sr. No.	Course Outcome	PO
1.		PO1, PO4
2.		PO1, PO6, PO10
3.		PO2, PO6, PO8
4.		PO2, PO4, PO11
5.		PO3, PO7, PO8, PO12
6.		PO1, PO5, PO9
7.		PO2, PO6, PO9, PO11

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
	√			√								
	√					√				√		
		√				√		√				
		√		√							√	
			√				√	√				√
	√				√				√			
		√					√		√			√

B. Tech. IV Semester (CSE)

OBJECT ORIENTED PROGRAMMING

L-T-P:/D Total	**Max Marks:**	**100**
3-0–0 5	**Final Exam:**	**70 Marks**
	Sessional:	**20 Marks**
	Internals:	**10 Marks.**

1. Introduction to C++: Object Oriented Technology, Advantages of OOP, Input- output in C++, Tokens, Keywords, Identifiers, Data Types C++, Derives data types. The void data type, Type Modifiers, Typecasting, Constant, Operator, Precedence of Operators, Strings.
Lecture: 3

2. Control Structures: Decision making statements like if-else, Nested if-else, goto, break, continue, switch case, Loop statement like for loop, nested for loop, while loop, do-while loop. Lecture: 3

3. Functions: Parts of Function, User- defined Functions, Value- Returning Functions, void Functions, Value Parameters, Function overloading, Virtual Functions. Lecture: 3

4. Classes and Data Abstraction: Structure in C++, Class, Build- in Operations on Classes, Assignment Operator and Classes, Class Scope, Reference parameters and Class Objects (Variables), Member functions, Accessor and Mutator Functions, Constructors, default Constructor, Destructors. Lecture: 15

5. Overloading & Templates: Operator Overloading, Function Overloading, Function Templates, Class Templates. Lecture: 5

6. Inheritance: Single and Multiple Inheritance, virtual Base class, Abstract Class, Pointer and Inheritance, Overloading Member Function. Lecture: 5

7. Pointers and Arrays: Void Pointers, Pointer to Class, Pointer to Object, This Pointer, Void Pointer, Arrays. Lecture: 6

8. Exception Handling: The keywords try, throw and catch. Creating own Exception Classes, Exception Handling Techniques (Terminate the Program, Fix the Error and Continue, Log the Error and Continue), Stack Unwinding. Lecture: 5

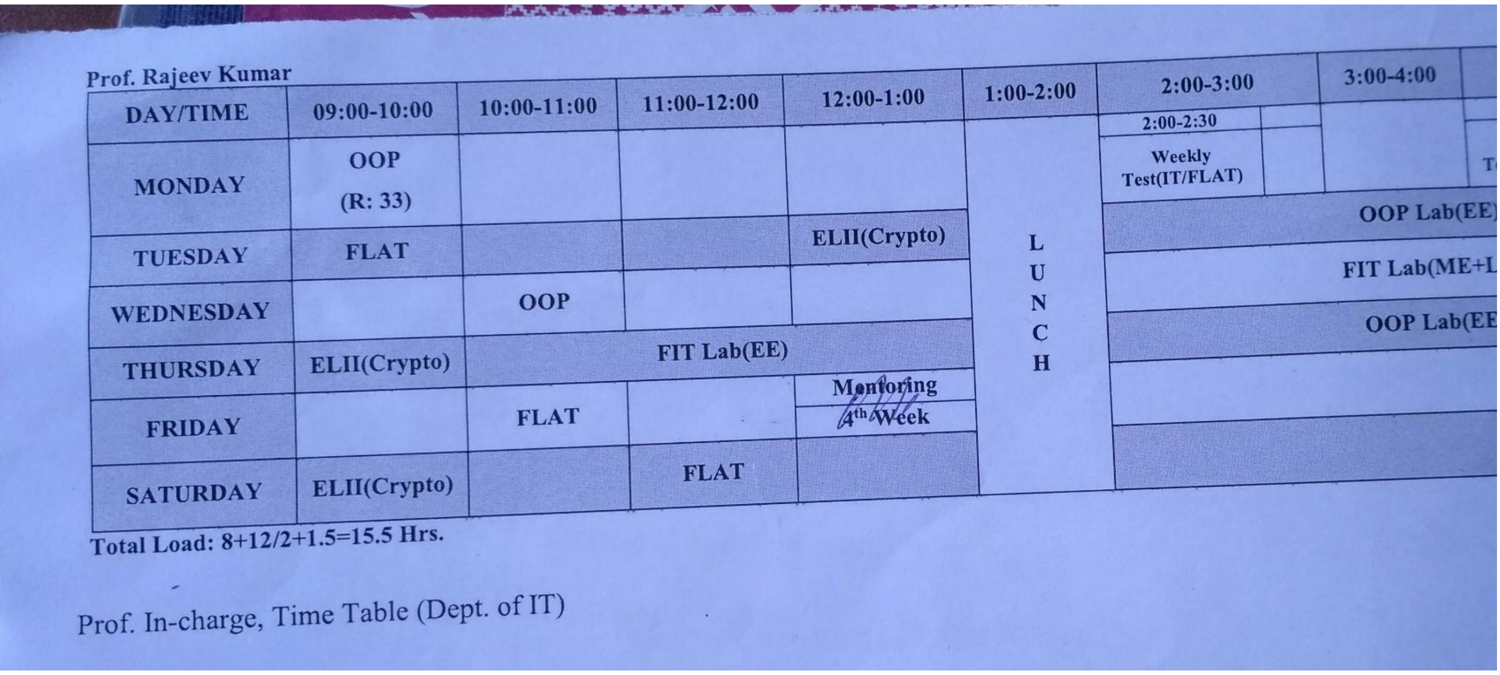
Text Books:

1. Thinking in C++, Volume 1 & 2 by Bruce Eckel, Chuck Allison, Pearson Education
2. Mastering C++, 1/e by Venugopal, Tata McGraw Hill.
3. Object Oriented Programming with C++, 3/e by E. Balaguruswamy, Tata McGraw Hill.
4. Starting Out with Object Oriented Programming in C++, by Tony Gaddis, Wiley India.

Reference Books:

1. The C++ Programming language 3/e by Bjarne Stroustrup, Pearson Education.
2. C++, How to Programme, 4e, by Deitel, Pearson Education.
3. Big C++ by Cay Horstmann, Wiley India.
4. C++ Primer, 3e by Stanley B. Lippmann, Josee Lajoie, Pearson Education.
5. C++ and Object Oriented Programming Paradigm, 2e by Debasish Jana, PHI.
6. Programming with C++, 2/e by Ravichandran, Tata McGraw Hill.
7. C++ Programming Black Book by Steven Holzner, Dreamtech Press.

Programming Lab (OOP) Writing programs using classes and objects, constructors and destructors, Inheritance properties, overloading operators, Use of pointers list representation, Programs and virtual functions, file handling, I. O. Manipulator using C++.



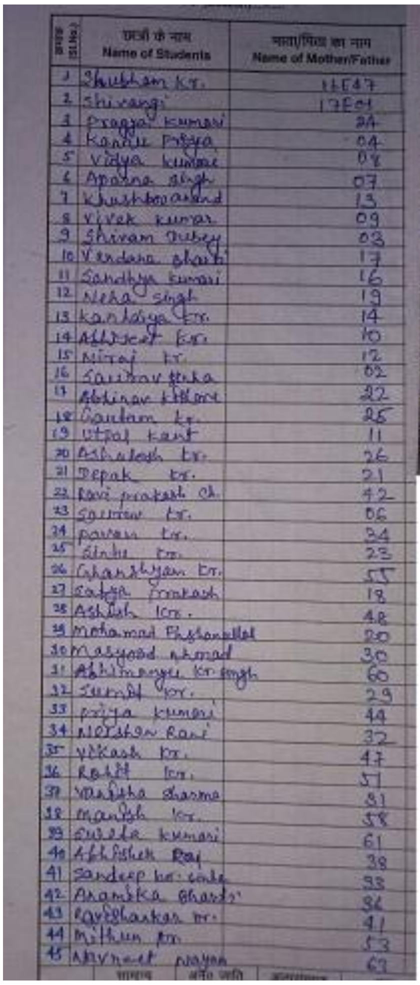
Prof. Rajeev Kumar

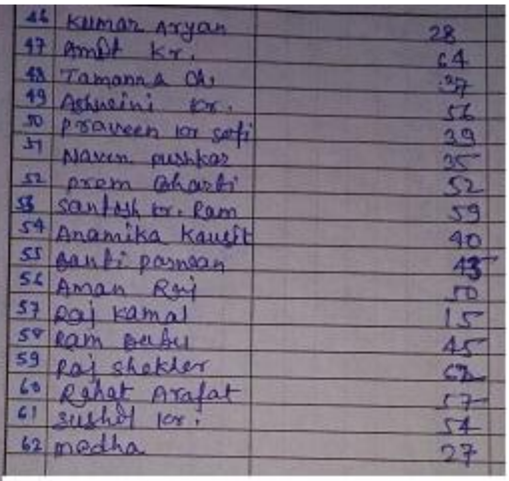
DAY/TIME	09:00-10:00	10:00-11:00	11:00-12:00	12:00-1:00	1:00-2:00	2:00-3:00		3:00-4:00	
						2:00-2:30			
MONDAY	OOP (R: 33)				L U N C H	Weekly Test(IT/FLAT)			T
TUESDAY	FLAT			ELII(Crypto)		OOP Lab(EE)			
WEDNESDAY		OOP				FIT Lab(ME+L			
THURSDAY	ELII(Crypto)	FIT Lab(EE)				OOP Lab(EE			
FRIDAY		FLAT		Mentoring 4th Week					
SATURDAY	ELII(Crypto)		FLAT						

Total Load: 8+12/2+1.5=15.5 Hrs.

Prof. In-charge, Time Table (Dept. of IT)

EE Student List(ROLL NO. WISE)





क्रमांक (Sl.No.)	छात्रों के नाम Name of Students	माता/पिता का नाम Name of Mother/Father
1	Shubham Kr.	16E47
2	shivangi	17E01
3	Pragya kumari	24
4	Kanchi Priya	04
5	Vidya kumari	08
6	Aparna singh	07
7	Khushboo anand	13
8	vivek kumar	09
9	Shivam Dubey	03
10	Vandana bharti	17
11	Sandhya kumari	16
12	Neha singh	19
13	kanhaiya kr.	14
14	Abhijeet kr.	10
15	Niraj kr.	12
16	Saurav Sinha	02
17	Abhinav Kishore	22
18	Gautam kr.	25
19	Utpal kant	11
20	Ashutosh kr.	26
21	Deepak kr.	21
22	Ravi prakash Ch.	42
23	Saurav kr.	06
24	pawan kr.	34
25	Sintu kr.	23
26	Ghanshyam kr.	55
27	Satya prakash	18
28	Ashish kr.	48
29	Mohamad Fazlanullah	20
30	Masyood Ahmad	30
31	Abhimanyu kr singh	60
32	Sumit kr.	29
33	priya kumari	44
34	Nirshan Rani	32
35	vikash kr.	47
36	Rohit kr.	57
37	Vashistha sharma	31
38	manish kr.	58
39	Sweta kumari	61
40	Abhishek Raj	38
41	Sandeep kr. sinha	33
42	Anamika Bharti	36
43	Ravishankar kr.	41
44	Mithun kr.	53
45	Abhinit narayan	63
46	Kumar Aryan	28
47	Amit kr.	64
48	Tamanna Ch.	37
49	Ashwini kr.	56
50	praveen kr safi	39
51	Naveen pushkar	35
52	prem bharti	52
53	santosh kr. Ram	59
54	Anamika Kaushik	40
55	Banti paswan	43
56	Aman Raj	50
57	Raj kamal	15
58	Ram babu	45
59	Raj shekher	62
60	Rahat Arafat	57
61	sushil kr.	54
62	medha	27

Institute / School Name :	Muzaffarpur Institute Of Technology, Muzaffarpur, Bihar		
Program Name	**EE B.Tech II (III SEM)**		
Course Code	051X01		
Course Name	OBJECT ORIENTED PROGRAMMING		
Lecture / Tutorial (per week):	2(2HR)	**Course Credits**	0
Course Coordinator Name	Assistant Professor Rajeev Kumar		

1. **Scope and Objectives of the Course**

   2. Perform object oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O. and other standard language constructs.
   3. Demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.
   4. Demonstrate ability to implement one or more patterns involving realization of an abstract interface and utilization of polymorphism in the solution of problems which can take advantage of dynamic dispatching.
   5. Learn syntax, features of, and how to utilize the Standard Template Library.
   6. To study the concepts of Assembler, Macro Processor, Loader and Linker

2. Text Books:

1. Thinking in C++, Volume 1 & 2 by Bruce Eckel, Chuck Allison, Pearson Education

2. Mastering C++, 1/e by Venugopal, Tata McGraw Hill.

3. Object Oriented Programming with C++, 3/e by E. Balaguruswamy, Tata McGraw Hill.

4. Starting Out with Object Oriented Programming in C++, by Tony Gaddis, Wiley India.

3. Reference Books:

1. The C++ Programming language 3/e by Bjarne Stroustrup, Pearson Education.

2. C++, How to Programme, 4e, by Deitel, Pearson Education.

3. Big C++ by Cay Horstmann, Wiley India.

4. C++ Primer, 3e by Stanley B. Lippmann, Josee Lajoie, Pearson Education.

5. C++ and Object Oriented Programming Paradigm, 2e by Debasish Jana, PHI.

6. Programming with C++, 2/e by Ravichandran, Tata McGraw Hill.

7. C++ Programming Black Book by Steven Holzner, Dreamtech Press.

1. **Evaluation Scheme:**

Component 1*	Sessional Test (ST)*	20
Component 2	Assignment Evaluation	10
Component 3**	End Term Examination**	70
	Total	**100**

Assignment

1. Define OOP? Write Down feature of OOPS.
2. Define Advantages of OOPS.
3. Define function and overloading of function.
4. Explain call by value, call by address, call by references with examples.
5. Explain control statements and various type of control statements?
6. Explain input and output in c++ with examples?
7. Explain class and object.
8. Explain member function with an example.
9. Write down friend function with example.
10. Write down friend function overloading with example.
11. Write down constructor and destructor and type of constructor with example.
12. Write down constructor overloading with example.
13. Difference between structure and class?
14. Define static members.
15. Write down binary operator and unary operator overloading with example.
16. Write down Inheritance and types of inheritance with example.
17. Write down polymorphism with example.
18. Write down this pointer with example.
19. Write down virtual function with example.
20. Write down abstract class with example.
21. Write down pointer to object and object to pointer with example.
22. Write down template and type of template with example.
23. Write down exception handling with example.

Quality Measurement Sheets

a. Course End Survey

ACADEMIC YEAR:	SEM:	DATE:
COURSE:	CLASS:	FACULTY:

Please evaluate on the following scale:

Excellent(E)	Good(G)	Average(A)	Poor(P)	No Comment(NC)
5	4	3	2	1

SNO	**QUESTIONAIRE**	**E**	**G**	**A**	**P**	**NC**	
		5	**4**	**3**	**2**	**1**	
GENERAL OBJECTIVES:							
1	Did the course achieve its stated objectives?						
2	Have you acquired the stated skills?						
3	Whether the syllabus content is adequate to achieve the objectives?						
4	Whether the instructor has helped you in acquiring the stated skills?						
5	Whether the instructor has given real life applications of the course?						
6	Whether tests, assignments, projects and grading were fair?						
7	The instructional approach (es) used was (were) appropriate to the course.						
8	The instructor motivated me to do my best work.						
9	I gave my best effort in this course						
10	To what extent you feel the course outcomes have been achieved.						
Please provide written comments:							

a) What was the most effective part of this course

b) What are your suggestions, if any, for changes that would improve this course?

c) Given all that you learned as a result of this course, what do you consider to be most import

d) Do you have any additional comments or clarifications to make regarding your responses tc any particular survey item?

e) Do you have any additional comments or suggestions that go beyond issues addressed on tl survey?