# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

Practical No: 9

**1) Write a Program to use all** the function of Introspection.

CODE:

<?php

class A

{

public string $name = 'Spectre';

public string $age = '19';

public function getData()

{

echo "getData()";

}

public function showData()

{

echo "showData()";

}

}

class B extends A

{

}

interface C

{

}

if (class\_exists('A')) {

echo "Class A exists";

} else {

echo "Class A doesnt exists";

}

echo "\n";

$obj = new B();

echo "Class Name: " . get\_class($obj);

echo "\n";

echo "Parent Class: " . get\_parent\_class($obj);

echo "\n";

if (is\_subclass\_of($obj, 'A')) {

echo "Object 'obj' is a subclass of class 'A'.";

} else {

echo "Object 'obj' is not a subclass of class 'A'.";

}

echo "\n";

$all\_classes = get\_declared\_classes();

print\_r($all\_classes);

echo "\n";

$my\_class\_methods = get\_class\_methods('A');

print\_r($my\_class\_methods);

echo "\n";

$my\_class\_props = get\_class\_vars('A');

print\_r($my\_class\_props);

echo "\n";

if (interface\_exists('C')){

echo "Interface C lives on...";

} else {

echo "Interface C died...";

}

echo "\n";

echo "\n";

if (method\_exists($obj, 'showData')) {

echo "Method 'showData' exists!";

} else {

echo "Method 'showData' does not exist!";

}

**OUTPUT:**

C:\xampp\php\php.exe C:\xampp\htdocs\WBP\_Documentation\WBPExperiment9.php

Class A exists

Class Name: B

Parent Class: A

Object 'obj' is a subclass of class 'A'.

Array

(

[0] => InternalIterator

[1] => Exception

[2] => ErrorException

[3] => Error

[4] => CompileError

[5] => ParseError

[6] => TypeError

[7] => ArgumentCountError

[8] => ValueError

[9] => ArithmeticError

[10] => DivisionByZeroError

[11] => UnhandledMatchError

[12] => Closure

[13] => Generator

[14] => ClosedGeneratorException

[15] => WeakReference

[16] => WeakMap

[17] => Attribute

[18] => ReturnTypeWillChange

[19] => AllowDynamicProperties

[20] => SensitiveParameter

[21] => SensitiveParameterValue

[22] => Fiber

[23] => FiberError

[24] => stdClass

[25] => DateTime

[26] => DateTimeImmutable

[27] => DateTimeZone

[28] => DateInterval

[29] => DatePeriod

[30] => HashContext

[31] => JsonException

[32] => LogicException

[33] => BadFunctionCallException

[34] => BadMethodCallException

[35] => DomainException

[36] => InvalidArgumentException

[37] => LengthException

[38] => OutOfRangeException

[39] => RuntimeException

[40] => OutOfBoundsException

[41] => OverflowException

[42] => RangeException

[43] => UnderflowException

[44] => UnexpectedValueException

[45] => RecursiveIteratorIterator

[46] => IteratorIterator

[47] => FilterIterator

[48] => RecursiveFilterIterator

[49] => CallbackFilterIterator

[50] => RecursiveCallbackFilterIterator

[51] => ParentIterator

[52] => LimitIterator

[53] => CachingIterator

[54] => RecursiveCachingIterator

[55] => NoRewindIterator

[56] => AppendIterator

[57] => InfiniteIterator

[58] => RegexIterator

[59] => RecursiveRegexIterator

[60] => EmptyIterator

[61] => RecursiveTreeIterator

[62] => ArrayObject

[63] => ArrayIterator

[64] => RecursiveArrayIterator

[65] => SplFileInfo

[66] => DirectoryIterator

[67] => FilesystemIterator

[68] => RecursiveDirectoryIterator

[69] => GlobIterator

[70] => SplFileObject

[71] => SplTempFileObject

[72] => SplDoublyLinkedList

[73] => SplQueue

[74] => SplStack

[75] => SplHeap

[76] => SplMinHeap

[77] => SplMaxHeap

[78] => SplPriorityQueue

[79] => SplFixedArray

[80] => SplObjectStorage

[81] => MultipleIterator

[82] => Random\RandomError

[83] => Random\BrokenRandomEngineError

[84] => Random\RandomException

[85] => Random\Engine\Mt19937

[86] => Random\Engine\PcgOneseq128XslRr64

[87] => Random\Engine\Xoshiro256StarStar

[88] => Random\Engine\Secure

[89] => Random\Randomizer

[90] => ReflectionException

[91] => Reflection

[92] => ReflectionFunctionAbstract

[93] => ReflectionFunction

[94] => ReflectionGenerator

[95] => ReflectionParameter

[96] => ReflectionType

[97] => ReflectionNamedType

[98] => ReflectionUnionType

[99] => ReflectionIntersectionType

[100] => ReflectionMethod

[101] => ReflectionClass

[102] => ReflectionObject

[103] => ReflectionProperty

[104] => ReflectionClassConstant

[105] => ReflectionExtension

[106] => ReflectionZendExtension

[107] => ReflectionReference

[108] => ReflectionAttribute

[109] => ReflectionEnum

[110] => ReflectionEnumUnitCase

[111] => ReflectionEnumBackedCase

[112] => ReflectionFiber

[113] => SessionHandler

[114] => \_\_PHP\_Incomplete\_Class

[115] => AssertionError

[116] => php\_user\_filter

[117] => Directory

[118] => PhpToken

[119] => InflateContext

[120] => DeflateContext

[121] => LibXMLError

[122] => DOMException

[123] => DOMImplementation

[124] => DOMNode

[125] => DOMNameSpaceNode

[126] => DOMDocumentFragment

[127] => DOMDocument

[128] => DOMNodeList

[129] => DOMNamedNodeMap

[130] => DOMCharacterData

[131] => DOMAttr

[132] => DOMElement

[133] => DOMText

[134] => DOMComment

[135] => DOMCdataSection

[136] => DOMDocumentType

[137] => DOMNotation

[138] => DOMEntity

[139] => DOMEntityReference

[140] => DOMProcessingInstruction

[141] => DOMXPath

[142] => PDOException

[143] => PDO

[144] => PDOStatement

[145] => PDORow

[146] => SimpleXMLElement

[147] => SimpleXMLIterator

[148] => XMLParser

[149] => XMLReader

[150] => XMLWriter

[151] => OpenSSLCertificate

[152] => OpenSSLCertificateSigningRequest

[153] => OpenSSLAsymmetricKey

[154] => CurlHandle

[155] => CurlMultiHandle

[156] => CurlShareHandle

[157] => CURLFile

[158] => CURLStringFile

[159] => finfo

[160] => mysqli\_sql\_exception

[161] => mysqli\_driver

[162] => mysqli

[163] => mysqli\_warning

[164] => mysqli\_result

[165] => mysqli\_stmt

[166] => PharException

[167] => Phar

[168] => PharData

[169] => PharFileInfo

[170] => FTP\Connection

[171] => A

[172] => B

)

Array

(

[0] => getData

[1] => showData

)

Array

(

[name] => Spectre

[age] => 19

)

Interface C lives on...

Method 'showData' exists!

Process finished with exit code

2) Write a Program to use the function of Serialization.

CODE:

class Game {  
 public string $name;  
 public string $char;  
  
 public function \_\_construct($name, $age) {  
 $this->name = $name;  
 $this->char = $age;  
 }  
}  
  
$person = new Game("Horizon", "Aloy");  
  
$serialized = serialize($person);  
  
echo "Serialized Object: " . $serialized . "\n\n";  
  
$deserialized = unserialize($serialized);  
  
echo "Deserialized Object: ";  
print\_r($deserialized);

OUTPUT:

